KU LEUVEN
FACULTEIT INGENIEURSWETENSCHAPPEN
DEPARTEMENT ELEKTROTECHNIEK
STADIUS CENTER FOR DYNAMICAL SYSTEMS,
SIGNAL PROCESSING AND DATA ANALYTICS
Kasteelpark Arenberg 10 – B-3001 Leuven

EMBEDDED OPTIMIZATION ALGORITHMS FOR PERCEPTUAL ENHANCEMENT OF AUDIO SIGNALS

Jury:
Prof. dr. C. Vandecasteele, voorzitter
Prof. dr. ir. P. Sas, vervangend voorzitter
Prof. dr. ir. M. Moonen, promotor
Prof. dr. M. Diehl, promotor
Prof. dr. ir. T. van Waterschoot, co-promotor
Prof. dr. ir. J. Suykens
Prof. dr. ir. P. Wambacq
Prof. dr. Y. Nesterov
(Université catholique de Louvain, Belgium)
Prof. dr. ir. W. Verhelst
(Vrije Universiteit Brussel, Belgium)

Proefschrift voorgedragen
tot het behalen van de
graad van Doctor in de
Ingenieurswetenschappen

door

Bruno DEFRAENE

December 2013

ISBN 978-94-6018-779-7

D/2013/7515/160

Voorwoord

Achteromkijkend hangt een levensloop aaneen van de toevalligheden. Als eerstejaars masterstudent elektrotechniek was het dan ook eerder toevallig dat ik medio april 2008 tijdens de ESAT-eindwerkbeurs in gesprek raakte met Toon van Waterschoot, die er de eindwerkvoorstellen van de DSP-groep aanprees. Een clipper ontwerpen voor audiosignalen? Klonk me heel interessant in de oren. Optimalisatie als voorgestelde aanpak voor het probleem? Vernieuwend idee! En dit alles onder een promotorenduo bestaande uit Prof. Marc Moonen en Prof. Moritz Diehl? Mijn eerste masterproefkeuze was beslist, hier wilde ik heel graag een academiejaar lang onderzoek naar doen. Die dag begon voor mij - zonder het zelf te beseffen - een wetenschappelijk en menselijk avontuur vol onvergetelijke momenten, inspirerende ontmoetingen, en verhelderende inzichten, een verrassende rondreis die me onder andere naar het Verre Oosten en de Maghreb zou brengen.

Het succesvol vervullen van mijn doctoraatsonderzoek heeft - veel meer dan met toeval - vooral te maken met de uitstekende begeleiders, het boeiende onderzoeksonderwerp, de fijne collega's, en natuurlijk de onmisbare en onvoorwaardelijke steun van het thuisfront, die er samen hebben voor gezorgd dat vier jaar hard werken konden culmineren in deze doctoraatstekst. In wat volgt zou ik graag alle mensen die me hierbij hebben geholpen oprecht willen bedanken.

Vooreerst zou ik mijn promotoren willen bedanken voor hun uitstekende begeleiding van mijn onderzoek. Prof. Marc Moonen wil ik bedanken om me de kans te geven om onder zijn promotorschap een doctoraat te maken in zijn - het mag gezegd - gerenommeerde onderzoeksgroep. Zijn uitstekende lessen over digitale (audio)-signaalverwerking hebben me reeds tijdens mijn ingenieursopleiding geboeid en gevormd. Marc, heel erg bedankt om doorheen het doctoraat in mij te geloven, voor de talloze ideeën die tijdens onze vrijdagse meetings zijn ontstaan, voor je visie om op het gepaste moment de onderzoeksrichting bij te sturen, voor je uitstekende correcties die mijn teksten stuk voor stuk publicatierijp hebben gemaakt, en voor de zin voor detail, structuur en kritiek die je me hebt bijgebracht.

Prof. Moritz Diehl zou ik willen bedanken om me als promotor met een ongeevenaard enthousiasme de weg te wijzen in het domein van de wiskundige optimalisatie. De vele onderzoeksideeën die hij gelanceerd heeft, de contacten die hij voor mij gelegd heeft met vooraanstaande optimalisatiespecialisten binnen en buiten OPTEC, en de steeds opbouwende kritische blik waarmee de eerste versie van publicaties werden doorgelicht, hebben in hoge mate bijgedragen aan de behaalde onderzoeksresultaten. Van harte bedankt hiervoor, Moritz.

Prof. Toon van Waterschoot wil ik graag bedanken voor de uitzonderlijk goede manier waarop hij zowel mijn masterproef en doctoraatsonderzoek heeft begeleid. Toon heeft me ingewijd in het verrichten van wetenschappelijk onderzoek, in het duidelijk synthetiseren en rapporteren, in het begeleiden van masterproefstudenten, en in ontelbare andere dingen waarin hij voor mij gedurende vijf jaar een inspirerende leermeester is geweest. Toon, ik heb enorm veel opgestoken van onze nauwe samenwerking, bedankt voor alles!

Ik zou ook de leden van de examencommissie hartelijk willen danken voor hun bereidheid om deel uit te maken van de jury, voor de kritische lezing van mijn proefschrift, en voor de interessante suggesties voor verbetering. *Dear Prof. Johan Suykens, Prof. Patrick Wambacq, Prof. Werner Verhelst. Prof. Yurii Nesterov, Prof. Carlo Vandecasteele, Prof. Paul Sas, I want to thank all of you for being part of my examination committee and for providing your valuable suggestions for the improvement of this manuscript.*

Ik heb tijdens mijn doctoraat de eer gehad om samen te mogen werken met verschillende onderzoekers die elk met hun eigen expertise een belangrijke contributie hebben geleverd aan de behaalde onderzoeksresultaten. *Dr. Hans Joachim Ferreau, thank you for introducing me to the art of QP solving in the early stages of my PhD. Dr. Andrea Suardi, thank you so much for your time and efforts spent at successfully implementing the clipping algorithm in hardware. Dr. Kim Ngo, thank you for our fruitful cooperation on the speech enhancement project.* Naim Mansour en Steven De Hertogh wil ik bedanken voor de vlotte samenwerking tijdens maar ook na het afleggen van hun uitmuntende masterproef.

Graag zou ik ook de collega's binnen de DSP-onderzoeksgroep willen bedanken, die samen gezorgd hebben voor een heel leuke en kameraadschappelijke sfeer, waarin het altijd prettig werken was. De fijne herinneringen aan mijn vier jaren in deze unieke groep zijn legio (en beperken zich niet tot binnen de muren van het departement): ik denk onwillekeurig aan het jaarlijkse ESAT-voetbaltornooi waarin deelnemen achteraf toch een pak belangrijker bleek dan winnen, aan de conferenties in dichtbije of avontuurlijk verre buitenlanden, aan de leuke etentjes, en natuurlijk aan de legendarische housewarmings en feestjes in het decor van de Leuvense binnenstad. Paschalis, bedankt om als vaste bureaugenoot altijd klaar te staan met goede raad, voor onze interessante gesprekken, voor het voortdurend delen van je inzichten en ervaring.

Alexander en Bram, bedankt om een lichtend voorbeeld te vormen van hoe je een doctoraat tot een (zeer) succesvol einde brengt, en natuurlijk ook voor de gezamenlijke Alma-bezoeken die steeds een zeer aangenaam rustpunt in de dag vormden.

Rodrigo, Joe, Javier, Pepe, Amir, thanks for all the nice moments we have shared within and outside of ESAT. Beier and Lulu, it was truly an honour for me to attend your wedding party in Beijing, the visit to China with Bram and Pieter was an unforgettable experience, for which I thank you with all of my heart. I would like to thank all my colleagues for creating such a nice work atmosphere throughout the years: thank you Aldona, Amin, Ann, Deepak, Enzo, Gert, Giacomo, Giuliano, Hanne, Johnny, Jorge, Kristian, Marijn, Nejem, Niccolo, Prabin, Rodolfo, Romain, Sylwek, Wouter, and Yi!

Ik zou ook een woord van dank willen richten aan de collega's die het departement ESAT en de afdeling STADIUS (formerly known as SISTA) al die jaren logistiek, organisatorisch en financieel vlot draaiende hebben gehouden: bedankt Ida, Lut, Eliane, Evelyn, en Ilse voor jullie harde werk. John zou ik daarnaast ook willen bedanken voor de vele momenten van gedeelde vreugde (vaak) en smart (héél soms) na de voetbalprestaties van ons geliefde RSC Anderlecht.

Mattia, I want to thank you for being an ever-enjoyable flatmate, I have truly appreciated our years of shared ups and downs in the quest for a successful PhD, as much as the memorable parties that were hosted in our apartment.

Ten slotte zou ik de mensen willen bedanken die me het dierbaarst zijn. Mama, papa, ik wil jullie hier oneindig bedanken voor de manier waarop jullie mij opgevoed hebben, voor jullie onvoorwaardelijke steun, geloof en interesse in alles wat ik doe, en voor alle goede raadgevingen die jullie me steeds weer hebben gegeven. Dit doctoraatsproefschrift tot een goed einde brengen zou zonder jullie onmogelijk zijn geweest. *Papa, maman, je tiens à vous remercier infiniment pour la façon dont vous m' avez élevé, et pour votre soutien inconditionnel dans tout ce que je fais. Il aurait été impossible de finir mon doctorat sans votre soutien.* Gilles, jou wil ik bedanken om als grote broer voor mij het pad te effenen en het goede voorbeeld te tonen als burgerlijk ingenieur, muzikant, en op vele andere vlakken, met het afwerken van dit proefschrift ben ik jou - voor de verandering - eens voorgegaan. Oma, bedankt voor de grote betrokkenheid die je samen met Parrain altijd getoond hebt in alle stappen die ik heb gezet, voor het goede voorbeeld dat jullie me steeds getoond hebben, en de wijze raad en ervaring die jullie mij hebben doorgegeven. *Marraine, merci beaucoup pour ta générosité, ton accueil toujours aussi chaleureux, et pour tout ce que tu m'as appris.* Mijn oprechte dank gaat ook uit naar Anne, Philippe, Lotte, Jasper, Paulien, Gaby, Agnès, en alle andere familieleden die me altijd gesteund hebben.

Lieve Sophie, jou wil ik danken voor je warme steun en liefde waarop ik in de laatste twee jaren altijd kon rekenen en die enorm veel voor mij betekenen, voor je onvoorwaardelijke geloof in mij, en voor alle prachtige momenten die we samen al hebben gedeeld.

Het naderende einde van dit voorwoord betekent voor sommigen misschien het sein om dit boek als gelezen te beschouwen. Diegenen die de voorgaande pagina's echter doorworsteld hebben om eindelijk aan het interessantere leeswerk te beginnen, wens ik naast proficiat ook veel leesplezier, in de hoop dat de verderop beschreven ideeën en onderzoeksresultaten een bouwsteen kunnen vormen voor nieuwe wetenschappelijke bevindingen.

Bruno Defraene

Leuven, December 2013

Abstract

This thesis investigates the design and evaluation of an embedded optimization framework for the perceptual enhancement of audio signals which are degraded by linear and/or nonlinear distortion. In general, audio signal enhancement has the goal to improve the perceived audio quality, speech intelligibility, or another desired perceptual attribute of the distorted audio signal by applying a real-time digital signal processing algorithm. In the designed embedded optimization framework, the audio signal enhancement problem under consideration is formulated and solved as a per-frame numerical optimization problem, allowing to compute the enhanced audio signal frame that is optimal according to a desired perceptual attribute. The first stage of the embedded optimization framework consists in the formulation of the per-frame optimization problem aimed at maximally enhancing the desired perceptual attribute, by explicitly incorporating a suitable model of human sound perception. The second stage of the embedded optimization framework consists in the on-line solution of the formulated per-frame optimization problem, by using a fast and reliable optimization method that exploits the inherent structure of the optimization problem. This embedded optimization framework is applied to four commonly encountered and challenging audio signal enhancement problems, namely hard clipping precompensation, loudspeaker precompensation, declipping and multi-microphone dereverberation.

The first part of this thesis focuses on *precompensation* algorithms, in which the audio signal enhancement operation is applied *before* the distortion process affects the audio signal. More specifically, the problems of hard clipping precompensation and loudspeaker precompensation are tackled in the embedded optimization framework. In the context of hard clipping precompensation, an objective function reflecting the perceptible nonlinear hard clipping distortion is constructed by including frequency weights based on the instantaneous masking threshold, which is computed on a frame-by frame basis by applying a perceptual model. The resulting per-frame convex quadratic optimization problems are solved efficiently using an optimal projected gradient method, for which theoretical complexity bounds are derived. Moreover, a fixed-point hardware implementation of this optimal projected gradient method on a field programmable gate array (FPGA) shows the algorithm to be capable to run in real time and without perceptible audio quality loss on a small and portable audio device. In the context of loudspeaker precompensation, an objective function reflecting the perceptible combined linear and nonlinear loudspeaker distortion is constructed in a similar fashion as for hard clipping precompensation. The loudspeaker is modeled using a Hammerstein loudspeaker model, i.e. a cascade of a memoryless nonlinearity and a linear FIR filter. The resulting per-frame nonconvex optimization problems are solved efficiently using gradient optimization methods which exploit knowledge on the invertibility and the smoothness of the memoryless nonlinearity in the Hammerstein loudspeaker model. From objective and subjective evaluation experiments, it is concluded with statistical significance that the embedded optimization algorithms for hard clipping and loudspeaker precompensation improve the resulting audio quality when compared to standard precompensation algorithms.

The second part of this thesis focuses on *recovery* algorithms, in which the audio signal enhancement operation is applied *after* the distortion process affects the audio signal. More specifically, the problems of declipping and multi-microphone dereverberation are tackled in the embedded optimization framework. Declipping is formulated as a sparse signal recovery problem where the recovery is performed by solving a per-frame ℓ_1-norm minimization problem, which includes frequency weights based on the instantaneous masking threshold. As a result, the declipping algorithm is focused on maximizing the perceived audio quality instead of the physical signal reconstruction quality of the declipped audio signal. Comparative objective and subjective evaluation experiments reveal with statistical significance that the proposed embedded optimization declipping algorithm improves the resulting audio quality compared to existing declipping algorithms. Multi-microphone dereverberation is formulated as a nonconvex optimization problem, allowing for the joint estimation of the clean audio signal and the room acoustics model parameters. It is shown that the nonconvex optimization problem can be smoothed by including regularization terms based on a statistical late reverberation model and a sparsity prior for the clean audio signal, which is demonstrated to improve the dereverberation performance.

Korte Inhoud

Dit doctoraatsproefschrift onderzoekt het ontwerp en de evaluatie van een ingebedde optimalisatieraamwerk voor de perceptuele verbetering van geluidssignalen die aangetast zijn door lineaire en niet-lineaire distortie. In het algemeen heeft signaalverbetering als doel om de geluidskwaliteit, spraakverstaanbaarheid, of een andere gewenste perceptuele eigenschap van het geluidssignaal te verbeteren door het toepassen van een digitaal signaalverwerkingsalgoritme in reële tijd. In het ontworpen ingebedde optimalisatieraamwerk wordt het beschouwde signaalverbeteringsprobleem geformuleerd en opgelost als een numeriek optimalisatieprobleem per signaalvenster, wat toelaat om het verbeterde signaalvenster te berekenen dat optimaal is volgens een gewenste perceptuele eigenschap. De eerste fase van het ingebedde optimalisatieraamwerk bestaat in de formulering van het optimalisatieprobleem per signaalvenster, en is erop gericht om de gewenste perceptuele eigenschap maximaal te verbeteren, door het toepassen van een geschikt model van de menselijke perceptie van geluid. De tweede fase van het ingebedde optimalisatieraamwerk bestaat in de online oplossing van het geformuleerde optimalisatieprobleem per signaalvenster, door het aanwenden van een snelle en betrouwbare optimalisatiemethode die de inherente structuur van het optimalisatieprobleem uitbuit. Dit ingebedde optimalisatieraamwerk wordt toegepast op vier courante en uitdagende signaalverbeteringsproblemen, namelijk de precompensatie van hard clipping, de precompensatie van luidsprekers, declipping, en meer-microfoons dereverberatie.

Het eerste deel van dit doctoraatsproefschrift spitst zich toe op algoritmes voor *signaalprecompensatie*, waarbij het geluidssignaal wordt verbeterd *voordat* de distortie inwerkt op het geluidssignaal. Meer specifiek worden de precompensatie van hard clipping en de precompensatie van luidsprekers als afzonderlijke problemen binnen het ingebedde optimalisatieraamwerk beschouwd. In het kader van de precompensatie van hard clipping, wordt een doelfunctie opgesteld die de waarneembare niet-lineaire hard clipping distortie weerspiegelt, door het toepassen van frequentiegewichten gebaseerd op de instantane maskeringsdrempel. Deze maskeringsdrempel wordt per signaalvenster berekend via een perceptueel model. Het resulterende convexe kwadratische

optimalisatieprobleem per signaalvenster wordt doeltreffend opgelost via een optimale geprojecteerde gradiëntmethode, waarvoor theoretische complexiteitsgrenzen worden opgesteld. Daarenboven toont een hardware implementatie in vaste komma van de optimale geprojecteerde gradiëntmethode op een field programmable gate array (FPGA) aan dat het algoritme in reële tijd en zonder waarneembaar geluidskwaliteitsverlies kan uitgevoerd worden op een klein en draagbaar audiotoestel. In het kader van de precompensatie van luidsprekers, wordt een doelfunctie opgesteld die de gecombineerde waarneembare lineaire en niet-lineaire luidsprekerdistortie weerspiegelt, op een gelijkaardige manier als voor de precompensatie van hard clipping. De luidspreker wordt gemodelleerd door een Hammerstein luidsprekermodel, dat bestaat uit de opeenvolging van een geheugenloze niet-lineariteit en een lineair FIR filter. Het resulterende niet-convexe optimalisatieprobleem per signaalvenster wordt doeltreffend opgelost via gradiëntmethodes die kennis uitbuiten over de inverteerbaarheid en gladheid van de geheugenloze niet-lineariteit in het Hammerstein luidsprekermodel. Objectieve en subjectieve evaluatie-experimenten laten toe om met statistische significantie te besluiten dat de ingebedde optimalisatiealgoritmes voor de precompensatie van hard clipping en luidsprekers de geluidskwaliteit verbeteren ten opzichte van bestaande algoritmes voor precompensatie.

Het tweede deel van dit doctoraatsproefschrift spitst zich toe op algoritmes voor *signaalherstel*, waarbij het geluidssignaal wordt verbeterd *nadat* de distortie heeft ingewerkt op het geluidssignaal. Meer bepaald worden declipping en meermicrofoons dereverberatie als afzonderlijke problemen binnen het ingebedde optimalisatieraamwerk beschouwd. Declipping wordt geformuleerd als een ijl signaalherstelprobleem, waarin het signaalherstel uitgevoerd wordt door het oplossen van een ℓ_1-norm minimalisatieprobleem per signaalvenster. Dit minimalisatieprobleem bevat frequentiegewichten gebaseerd op de instantane maskeringsdrempel. Zodoende poogt het declipping algoritme de geluidskwaliteit maximaal te verbeteren, in plaats van te focussen op de fysieke reconstructiekwaliteit van het geluidssignaal. Vergelijkende objectieve en subjectieve evaluatie-experimenten laten toe om met statistische significantie te besluiten dat het ingebedde optimalisatiealgoritme voor declipping de geluidskwaliteit verbetert ten opzichte van bestaande algoritmes. Meermicrofoons dereverberatie wordt geformuleerd als een niet-convex optimalisatieprobleem dat toelaat om gelijktijdig het zuivere geluidssignaal en de parameters van de kamerakoestiek te schatten. Het niet-convexe optimalisatieprobleem kan verzacht worden door regularisatietermen toe te voegen die gebaseerd zijn op een statistisch model voor late reverberatie en een ijlheidsveronderstelling van het zuivere geluidssignaal, die samen de performantie van dereverberatie aantoonbaar verhogen.

Glossary

Mathematical Notation

$\forall$	for all
$\triangleq$	defined as
$\cup$	set union
$\emptyset$	empty set
$\lVert\cdot\rVert$, $\lVert\cdot\rVert_p$	Euclidean vector norm, ℓ_p-norm
$(\cdot)^T$	matrix transpose
$(\cdot)^H$	Hermitian matrix transpose
$(\cdot)^{-1}$	matrix inverse
$(\cdot)^{+}$	Moore-Penrose pseudoinverse
$(\tilde{\cdot})$	median operator
$(\bar{\cdot})$	mean operator
$\mathrm{sgn}(\cdot)$	sign function
$\tanh(\cdot)$	hyperbolic tangent function
$\mathrm{diag}(\cdot)$	diagonal matrix operator
$\log_x(\cdot)$	logarithm in base x
$\max_x$	maximize over x
$\min_x$	minimize over x
$\inf_x$	infimum over x
$\mathbf{0}$	all zeros vector
$\mathbb{1}$	all ones vector
$\mathbb{N}$	set of natural numbers
$\mathbb{R}$	set of real numbers
$\mathbb{R}_+$	set of positive real numbers
$\mathbb{R}^N$	set of real N-dimensional vectors
$\mathbb{R}^{N\times N}$	set of real $N \times N$ matrices
$\mathbb{C}$	set of complex numbers
$\mathbb{C}^N$	set of complex N-dimensional vectors
$\mathbb{C}^{N\times N}$	set of complex $N \times N$ matrices

$\nabla(\cdot)$	gradient operator
$\nabla^2(\cdot)$	Hessian operator
$\otimes$	Kronecker product

Fixed Symbols

$\mathbf{a}_i$	sensing matrix column
$\mathbf{A}$	sensing matrix
$\mathbf{A}_m$	loudspeaker precompensation Hessian matrix
b	number of fraction bits
b_i	fraction bit
$\mathbf{b}_m$	loudspeaker precompensation gradient vector
c_{iter}	latency per iteration in clock cycles
c_{total}	overall latency in clock cycles
$\mathbf{c}_m^k$	auxiliary audio signal frame iterate
C_m	Lipschitz constant
$\mathbf{C}_m^+$	row selection matrix corresponding to positively clipped samples
$\mathbf{C}_m^-$	row selection matrix corresponding to negatively clipped samples
d_m	distance measure
d^*	optimal dual objective value
D	function domain
$\mathbf{D}$	unitary DFT matrix
e	Euler's number
e_i	decimal exponent bit
$\mathbf{e}$	error signal vector
E	decimal exponent
$E[\cdot]$	expected value operator
$f(\cdot)$	objective function
$\mathbf{f}(\cdot)$	distortion process
$g(\cdot)$	per-sample memoryless nonlinearity
$\mathbf{g}(\cdot)$	per-frame memoryless nonlinearity
$g^{-1}(\cdot)$	inverse per-sample memoryless nonlinearity
$\mathbf{g}^{-1}(\cdot)$	inverse per-frame memoryless nonlinearity
$h[n]$	finite impulse response
$\mathbf{h}$	RIR vector
H_0	statistical null hypothesis
H_1	statistical alternative hypothesis (Ch. 2)
H_a	statistical alternative hypothesis (Ch. 4,6)
$\mathbf{H}_0$	RIR matrix

$\mathbf{H}_m$	clipping precompensation Hessian matrix (Ch. 2)
	lower triangular convolution matrix (Ch. 3)
$\tilde{\mathbf{H}}_m$	upper triangular convolution matrix
i	discrete frequency index
$\mathbf{I}$	identity matrix
j	unit imaginary number
k	discrete iteration index
	sparsity (Ch. 6)
k'	approximate sparsity
K	fixed number of iterations
$K_{\max}$	maximum number of iterations
L	lower clipping level (Ch. 2)
	FIR filter order (Ch. 3)
$\mathbf{l}$	lower clipping level vector
m	discrete frame index (Ch. 1-6)
	microphone index (Ch. 7)
M	mantissa (Ch. 5)
	measurement length (Ch. 6)
	number of microphones (Ch. 7)
n	discrete sample index
N	frame length
N_{ps}	number of stimuli pairs
$O(\cdot)$	Landau symbol
p^*	optimal primal objective value
P	overlap length
$\mathbf{P}_m$	perceptual weighting matrix
$q(\cdot)$	Lagrange dual function
Q	convex feasible set
$\mathbf{Q}_m$	reduced loudspeaker precompensation Hessian matrix
r	amplitude level parameter for hyperbolic tangent function
s_0	sign bit
s_m^k	stepsize
$\mathbf{s}$	original audio signal
$\mathbf{s}_0$	source signal vector
$\mathbb{S}_m^k$	set of active constraints
t	test statistic (Ch. 4)
	time index (Ch. 7)
$\mathbf{t}_m$	instantaneous global masking threshold
U	upper clipping level
$\mathbf{u}$	upper clipping level vector

$\mathbf{v}$	precompensated audio signal
$\mathbf{v}_m$	precompensated audio signal frame
V_{fix}	fixed point value
V_{float}	floating point value
$\mathbb{V}_m^k$	set of violated constraints
$\mathbf{w}_m$	perceptual weighting function
$\mathbf{W}_m$	perceptual weighting matrix
$x[n]$	discrete time-domain clean audio signal
$X_m(e^{j\omega_i})$	discrete frequency-domain clean audio signal
$\mathbf{x}$	clean audio signal
$\mathbf{x}_m$	clean audio signal frame
$y[n]$	discrete time-domain distorted audio signal
$Y_m(e^{j\omega_i})$	discrete frequency-domain distorted audio signal
$\mathbf{y}$	distorted audio signal
$\mathbf{y}_m$	distorted audio signal frame
$\mathbf{y}_m^k$	distorted audio signal frame iterate
$y^*[n]$	discrete time-domain enhanced audio signal
$\mathbf{y}^*$	enhanced audio signal
$\mathbf{y}_m^*$	enhanced audio signal frame
α	compression parameter
α_{JB}	significance level for Jarque-Bera statistical normality test
α_{TT}	significance level for statistical t-test
β	relaxation of the gradient for Armijo condition
$\beta_i(\cdot)$	eigenvalue operator
γ_m	regularization parameter
γ_m^k	optimal projected gradient method auxiliary weight
δ_m^k	optimal projected gradient method weight
$\boldsymbol{\delta}$	optimal projected gradient method weight vector
ϵ	solution accuracy
ϵ_m	relaxation parameter
η	fixed number of iterations (Ch. 2)
	backtracking factor for Armijo line search (Ch. 3)
θ_c	clipping level
$\boldsymbol{\theta}$	distortion model parameters
$\hat{\boldsymbol{\theta}}$	estimated distortion model parameters
κ_m	condition number
$\lambda_i(\cdot)$	eigenvalue operator
λ	Lagrange multiplier associated to inequality constraint
$\boldsymbol{\lambda}$	Lagrange multiplier vector associated to inequality constraints

$\boldsymbol{\lambda}_{m,l}$	Lagrange multiplier vector associated to lower clipping level constraints
$\boldsymbol{\lambda}_{m,u}$	Lagrange multiplier vector associated to upper clipping level constraints
$\mu(\cdot)$	coherence measure
μ_m	convexity parameter
ν	Lagrange multiplier associated to equality constraint
$\boldsymbol{\nu}$	Lagrange multiplier vector associated to equality constraints
π	Archimedes' constant
$\Pi_Q(\cdot)$	orthogonal projection onto set Q
ρ	population Pearson correlation coefficient
$\hat{\rho}$	sample Pearson correlation coefficient
$\sigma_i(\cdot)$	singular value operator
$\sum_{n=1}^{N}$	summation operator
$\boldsymbol{\Phi}$	measurement matrix
$\boldsymbol{\Psi}$	fixed basis
ω_i	discrete frequency variable
Ω	convex feasible set
$\mathcal{L}(\cdot)$	Lagrangian function

Acronyms and Abbreviations

ADC	analog-to-digital converter
BCD	block coordinate descent
BP	Basis Pursuit
CCR	Comparison Category Rating
CD	compact disc
CF	clipping factor
CLB	Configurable Logic Block
CS	Compressed Sensing
CSL0	CS-based declipping using ℓ_0-norm optimization
CSL1	CS-based declipping using ℓ_1-norm optimization
DAC	digital-to-analog converter
dB	decibel
DCR	Degradation Category Rating
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DSP	digital signal processing
e.g.	*exempli gratia*: for example
FF	flip-flop
FFT	Fast Fourier Transform

FIR	Finite Impulse Response
FPGA	field programmable gate array
HLS	high-level synthesis
Hz	hertz
i.e.	*id est*: that is
IFFT	Inverse Fast Fourier Transform
IIR	Infinite Impulse Response
IP	Intellectual Property
kHz	kilohertz
ℓ_1/ℓ_2-RNLS	NLS with ℓ_1-norm and ℓ_2-norm regularization
ℓ_2-RNLS	NLS with ℓ_2-norm regularization
LAB	Logic Array Block
LUT	lookup table
MDCR	Mean Degradation Category Rating
MHz	megahertz
ms	milliseconds
MSE	mean-squared error
mW	milliwatt
NLS	nonlinear least squares
ODG	Objective Difference Grade
OMP	Orthogonal Matching Pursuit
PCS	Perceptual Compressed Sensing
PCSL1	PCS-based declipping using ℓ_1-norm optimization
PEAQ	Perceptual Evaluation of Audio Quality
PSD	power spectral density
QP	quadratic program
RIP	restricted isometry property
RIR	room impulse response
RTL	register-transfer level
s	seconds
s.t.	subject to
SCP	sequential cone programming
SNR	signal-to-noise ratio
SPL	Sound Pressure Level
SQP	sequential quadratic programming
VHDL	VHSIC Hardware Description Language
VHSIC	very-high-speed integrated circuits
W	Watt
XPE	Xilinx Power Estimator
μs	microseconds

Contents

Voorwoord **i**

Abstract **v**

Korte Inhoud **vii**

Glossary **ix**

Contents **xvii**

I Introduction

1 Introduction **3**

1.1 Problem Statement and Motivation 4

1.1.1 Audio Signal Distortion 4

1.1.2 Impact on Sound Perception 7

1.1.3 Audio Signal Enhancement 9

1.2 Precompensation Algorithms 11

1.2.1 Hard Clipping Precompensation 11

1.2.2 Loudspeaker Precompensation 14

1.3 Recovery Algorithms 17

1.3.1 Declipping 17

1.3.2 Dereverberation 19

1.4 Embedded Optimization Framework for Audio Signal Enhancement 21

1.4.1 Embedded Optimization 22

1.4.2 Perceptual Models 23

1.4.3 Main Research Objectives 26

1.5 Thesis Outline and Publications 26

1.5.1 Chapter-By-Chapter Outline and Contributions 26

1.5.2 Included Publications 29

Bibliography 30

II Precompensation Algorithms

2 Hard Clipping Precompensation **43**

2.1 Introduction 45

2.2 Perception-Based Clipping 48

2.2.1 General Description of the Algorithm 48

2.2.2 Optimization Problem Formulation 49

2.2.3 Perceptual Weighting Function 50

2.3 Optimization Methods 53

2.3.1 Convex Optimization Framework 54

2.3.2 Dual Active Set Strategy 56

2.3.3 Projected Gradient Descent 59

2.3.4 Optimal Projected Gradient Descent 63

2.4 Simulation Results 66

2.4.1 Comparative Evaluation of Perceived Audio Quality . . 66

2.4.2 Experimental Evaluation of Algorithmic Complexity . . 72

2.4.3 Applicability in Real-Time Context 73

2.5 Conclusions 75

Bibliography 76

3 Loudspeaker Precompensation 79

3.1 Introduction 81

3.2 Embedded-Optimization-Based Precompensation 82

3.2.1 Hammerstein Model Description 82

3.2.2 Embedded-Optimization-Based Precompensation 84

3.2.3 Perceptual Weighting Function 86

3.3 Optimization Methods 88

3.3.1 Classes of Memoryless Nonlinearities 88

3.3.2 Invertible Memoryless Nonlinearities 89

3.3.3 Non-Invertible Smooth Memoryless Nonlinearities . . . 91

3.3.4 Non-Invertible Hard Clipping Memoryless Nonlinearities 92

3.3.5 Algorithmic Complexity Bounds 96

3.4 Audio Quality Evaluation 97

3.4.1 Synthetic Hammerstein Loudspeaker Models 97

3.4.2 Identified Hammerstein Loudspeaker Models 100

3.4.3 Identification of Hammerstein Model Parameters 104

3.5 Conclusions 105

Bibliography 105

4 Subjective Audio Quality Evaluation 109

4.1 Introduction 109

4.2 Research Questions and Hypotheses 110

4.3 Experimental Design and Set-up 112

4.4 Results and Statistical Analysis 116

4.4.1 Test Subject Responses 116

4.4.2 Statistical Hypothesis Testing 116

4.4.3 Correlation Between Subjective and Objective Scores . 118

4.5 Conclusions 119

Bibliography 121

5 Embedded Hardware Implementation 123

5.1 Introduction 123

5.2 Embedded Hardware Architecture 124

5.2.1 Optimal Projected Gradient Algorithm 124

5.2.2 FPGA Implementation Architecture 125

5.3 FPGA Implementation Aspects 127

5.3.1 Floating-Point and Fixed-Point Arithmetic 127

5.3.2 Fast Fourier Transform 130

5.4 Simulation Results 132

5.4.1 Simulation Set-up 132

5.4.2 Accuracy in Fixed-Point Arithmetic 133

5.4.3 Latency, Resource Usage and Power Consumption . . . 134

5.5 Conclusions 139

Bibliography 139

III Recovery Algorithms

6 Declipping Using Perceptual Compressed Sensing 143

6.1 Introduction 145

6.2 A CS Framework for Declipping . . . 146

6.2.1 CS Basic Principles . . . 147

6.2.2 Perfect Recovery Guarantees . . . 148

6.2.3 CS-Based Declipping . . . 149

6.3 A PCS Framework for Declipping . . . 152

6.3.1 Perceptual CS Framework . . . 152

6.3.2 Masking Threshold Calculation . . . 154

6.3.3 PCS-Based Declipping Using ℓ_1-norm Optimization . . 155

6.4 Evaluation . . . 157

6.4.1 Objective Evaluation . . . 157

6.4.2 Impact of Regularisation Parameter γ_m . . . 163

6.4.3 Impact of Masking Threshold Estimation Procedure . . 163

6.4.4 Subjective Evaluation . . . 164

6.4.5 Suitability of PEAQ ODG as Objective Measure . . . 169

6.5 Conclusions . . . 169

Bibliography . . . 170

7 Multi-Microphone Dereverberation 175

7.1 Introduction . . . 177

7.2 Problem Statement . . . 178

7.3 Embedded Optimization Algorithms . . . 180

7.3.1 NLS problem . . . 181

7.3.2 $\boldsymbol{\ell_2}$-regularized NLS problem . . . 181

7.3.3 $\boldsymbol{\ell_1/\ell_2}$-regularized NLS problem . . . 182

7.4 Evaluation . . . 183

7.5 Conclusions . . . 184

Bibliography . . . 185

8 Conclusions and Suggestions for Future Research 187

8.1 Summary and Conclusions 187

8.2 Suggestions for Future Research 190

Bibliography 192

Publication List 193

Curriculum Vitae 197

Part I

Introduction

Chapter 1

Introduction

When listening to music through a portable music player, a laptop, or a public address system, sound quality and clarity are crucial factors to make it an enjoyable experience. When hearing the voice of the person you are speaking to through a mobile phone, a teleconferencing system, or a hearing aid, the quality and intelligibility of the speech are decisive for a satisfactory and effective communication. Before reaching the ear, music and speech signals have passed through many stages in the so-called *audio signal path*, e.g. from the recording device over the transmission channel to a reproduction device. Throughout this audio signal path, there is an abundance of potential *audio signal distortion* mechanisms, which can have a negative effect on the quality and intelligibility of the perceived audio signal. This makes it indispensable to design and apply effective *audio signal enhancement* algorithms for improving the quality or intelligibility of audio signals that are degraded by a given distortion process, by applying some form of real-time digital signal processing.

This introduction is organized as follows. In Section 1.1, the major distortion mechanisms along a typical audio signal path will be pointed out and their impact on sound perception will be discussed. In Sections 1.2 and 1.3, the state of the art and the prevailing challenges for audio signal enhancement algorithms will be reviewed. In Section 1.4, a novel audio signal enhancement framework for overcoming the limitations of existing audio signal enhancement algorithms is outlined, which is based on the application of *embedded optimization* and *perceptual models*. The design of this embedded optimization framework and its application to different audio signal enhancement problems form the topic of this thesis.

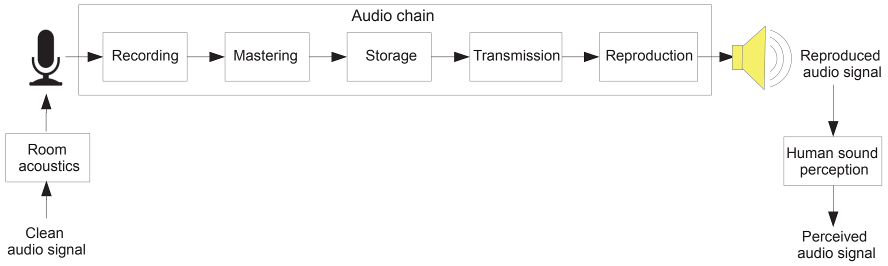


Figure 1.1: Stages in the audio signal path.

1.1 Problem Statement and Motivation

1.1.1 Audio Signal Distortion

Audio signal distortion can be defined as any alteration occurring in the time-domain waveform or frequency spectrum of an audio signal. Although in certain cases the distortion is applied intentionally to create a desired audio effect such as a distorted guitar sound, vocal reverberation, or a change of the audio signal timbre [1], in general the occurence of audio signal distortion is unintentional and undesired. Audio signal distortion can be broadly classified into two types, namely *linear* distortion and *nonlinear* distortion. Linear distortion involves changes in the relative amplitudes and phases of the frequency components constituting the original audio signal. Nonlinear distortion involves the introduction of frequency components that were not present in the original audio signal [2].

Linear and nonlinear distortion can be introduced at different stages along the audio signal path transforming the clean audio signal into the reproduced audio signal, as shown in Figure 1.1[1].

Room acoustics

In a first stage, the *room acoustics* form a potential source of audio signal distortion. When the clean audio signal is produced in a closed acoustic environment, it is partially reflected by the phyisical boundaries of the environment, i.e. by the walls, the floor, and ceiling of the room. As a result, not only the clean audio signal is picked up by the recording device, but also several delayed and attenuated replicas of the clean audio signal. This effect is known as *reverberation* and is a form of linear distortion [3].

[1] Note that not all stages in the audio signal path are necessarily present in all audio applications.

Recording

In a second stage, the *recording* device can cause additional audio signal distortion, due to non-idealities in the microphone and the subsequent analog-to-digital converter (ADC) or due to an incorrect microphone placement. At normal sound pressure levels, microphones typically have a non-uniform frequency response and phase response, leading to a linear distortion of the recorded signal. Moreover, at high sound pressure levels, the microphone will add nonlinear distortion due to several possible causes, such as nonlinear diaphragm motion or electrical overload of the internal amplifier and ADC [4][5].

Mastering

In a third stage, the recorded audio signal is prepared for storage on an analog or digital device through the application of a *mastering* process. The mastering stage involves dynamics processing using compressors, expanders and limiters for increasing or decreasing the dynamic range, and equalizing filters or bass boost filters for adjusting the spectral balance of the audio signal [6]. Although the mastering is applied intentionally to the audio signal, it is very common that undesired nonlinear distortion is unintentionally introduced, mainly due to the application of hypercompression and clipping in the quest for maximum loudness [7].

Storage

A fourth stage consists of the *storage* of the audio signal on an analog or digital storage device. Commonly used digital audio storage devices comprise magnetic devices (e.g. DAT, ADAT), optical devices (e.g. Compact Disc (CD), Super Audio CD (SACD), DVD, Blu-ray Disc (BD)), hard disks (e.g. on computers, USB, memory cards) and volatile memory devices. Commonly used analog audio storage devices comprise long playing vinyl records (LPs). In case a lossy audio codec is employed prior to storage on a digital device, compression artefacts include predominantly nonlinear distortion effects such as spectral valleys, spectral clipping, noise amplification, time-domain aliasing and tone trembling [8] [9]. Moreover, audio signal distortion can be introduced due to imperfections during the writing of the audio signal to the analog or digital storage device, or during the transcription between storage devices. As opposed to digital devices [10], analog audio storage devices are furthermore known to be very sensitive to wear and tear of the device itself, which can introduce considerable audio signal distortion [11].

Transmission

A fifth stage consists of the *transmission* of the stored audio signal through a wired or wireless communication network. Wireless transmission of audio signals is performed through analog radio broadcasting systems using amplitude modulation (AM) or frequency modulation (FM) technology, through digital radio broadcasting systems using Digital Audio Broadcasting (DAB) technology, or through mobile phone networks [4]. Wired transmission of audio signals is performed through Digital Subscriber Line (DSL), coaxial cable or optical fiber technology. Moreover, the recent proliferation of the Voice over Internet Protocol (VoIP) facilitates the delivery of voice communications over Internet Protocol (IP) networks. As analog transmission channels typically have reduced bandwith constraints and non-flat frequency responses, the introduction of linear distortion in the received audio signal is common. Moreover, in actual circumstances, wired or wireless digital transmission channels can not be regarded as error-free, meaning that they can be quantified by a nonzero bit error rate or packet error rate of the received data stream [12]. In general, these bit errors and packet errors can result in the introduction of nonlinear distortion and/or missing fragments in the received audio signal.

Reproduction

A sixth and last stage deals with the *reproduction* of the audio signal. Different aspects of sound reproduction can have an influence on the reproduced audio signal: the properties of the listening room, the digital-to-analog converter (DAC), the amplifier, and most dominantly the placement and properties of the loudspeaker system [13][14]. In general, loudspeakers have a non-ideal response introducing both linear and nonlinear distortion in the reproduced audio signal. At low amplitudes, the loudspeaker behaviour is almost linear and nonlinear signal distortion is negligible. However, at higher amplitudes nonlinear distortion occurs, the severity of which is correlated with the cost, weight, volume, and efficiency of the loudspeaker driver [15].

A wide variety of inherently nonlinear mechanisms are occurring in loudspeaker systems and are responsible for nonlinear distortion in the reproduced audio signal. The dominant nonlinear loudspeaker mechanisms are the following [16]:

- the nonlinear relation between the restoring force of the suspension and the voice coil displacement, due to the dependence of the stiffness of the suspension on the voice coil displacement;
- the nonlinear relation between the electro-dynamic driving force and the voice coil displacement, due to the dependence of the force factor on the voice coil displacement;
- the nonlinear relation between the electrical input impedance and the voice coil displacement, due to the dependence of the voice coil inductance

on the voice coil displacement;
- the nonlinear relation between the electrical input impedance and the electric input current, due to the dependence of the voice coil inductance on the electric input current.

Throughout the audio signal path, there are obviously many stages that can potentially add linear and nonlinear distortion to the clean audio signal, resulting in a reproduced audio signal that has an altered time-domain waveform and frequency spectrum compared to the clean audio signal. This audio signal distortion can have a significant impact on the perception of the audio signal by the listener, as will be discussed next.

1.1.2 Impact on Sound Perception

Depending on the application, the reproduced audio signal will be perceived by a human listener (e.g. in music playback systems, public address systems, voice communications, hearing assistance) or by a machine (e.g. in automatic speech recognition, music recognition/transcription). The focus in this thesis will be on human sound perception, but we should note that mitigating the effects of signal distortion on automatic speech [17] and music [18] recognition performance are active research topics as well.

The human perception of sound is a complex process involving both auditory and cognitive mechanisms. The resulting sound perception can be quantified using different perceptual attributes, depending on the nature of the audio signal and the application.

- For music signals, the perceived *audio quality* is the most important global perceptual attribute for the listener. The measurement of audio quality is a multidimensional problem that includes a number of individual perceptual attributes such as ‘clarity’, ‘loudness’, ‘sharpness’, ‘brightness’, ‘fullness’, ‘nearness’ and ‘spaciousness’ [19][20].
- For speech signals, the perceived *speech quality* and *speech intelligibility* are the most important global perceptual attributes for the listener. Speech quality also has a number of individual perceptual attributes, including ‘clarity’, ‘naturalness’, ‘loudness’, ‘listening effort’, ‘nasality’ and ‘graveness’ [21]. In the specific scenario of narrow-band and wideband telephone speech transmission, the perceptual attributes ‘discontinuity’, ‘noisiness’, ‘coloration’ and ‘loudness’ have been found to constitute speech quality [22][23]. Speech intelligibility in turn refers to how well the content of the speech signal can be identified by the listener, and is the primary concern in hearing aids and many speech communication systems. It is directly measurable by defining the proportion of speech items (e.g. syllables, words, sentences) that are correctly understood by the listener for a given speech intelligibility test [24].

Different listening experiments have been performed in order to assess the impact of linear and nonlinear audio signal distortion on the resulting audio quality, speech quality and speech intelligibility. The main results of these research efforts will be synthesized here.

Impact of Linear Distortion

Linear distortion is typically perceived as changing the timbre or coloration of the audio signal. The presence of linear distortion has been found to significantly affect the perceived quality of music and speech signals. It was experimentally shown that applying a linear filter possessing increasing frequency response irregularities (spectral tilts and ripples) or bandwidth restrictions (lower and upper cut-off frequency) results in an increasing degradation of the global perceived audio quality and speech quality [25]. Moreover, all the individual perceptual attributes constituting audio quality were found to be significantly affected by changing the frequency response [26]. On the other hand, the effects of changes in phase response were found to be generally small compared to the effects of irregularities in frequency magnitude response [27].

Linear distortion caused by reverberation is known to add spaciousness and coloration to the sound. For music signals, this is not necessarily an undesired property, however, for speech signals, reverberation is known to have a significant negative impact on both speech quality and speech intelligibility [28][29].

Impact of Nonlinear Distortion

Nonlinear distortion is typically perceived as adding harshness or noisiness, or as the perception of sounds that were not present in the original signal, such as crackles or clicks. The presence of nonlinear distortion has been found to result in a significant degradation of the perceived quality of music and speech signals, both when artificial nonlinear distortions (e.g. hard clipping, soft clipping) and nonlinear distortions occurring in real transducers are considered [2]. In another experimental study, speech quality ratings for speech fragments exhibiting nonlinear hard clipping distortion have been found to decrease monotonically with increasing signal distortion, both for normal-hearing and hearing-impaired subjects [30]. Moreover, through speech intelligibility tests, it has been concluded that nonlinear distortion reduces speech intelligibility, both for normal-hearing and hearing-impaired listeners. For all listeners, the speech intelligibility scores were seen to decrease as the amount of nonlinear clipping distortion was increased [31].

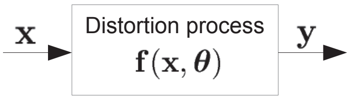


Figure 1.2: Audio signal distortion process.

Impact of Combined Linear and Nonlinear Distortion

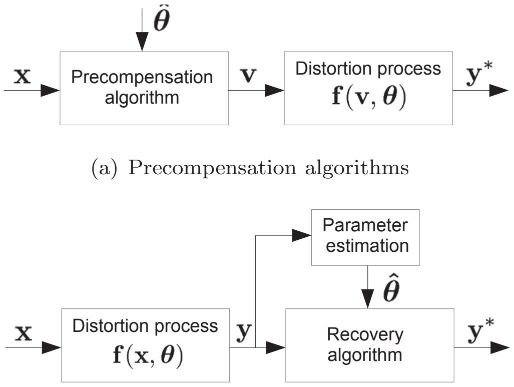
The impact of the simultaneous presence of linear distortion and nonlinear distortion has been studied in listening experiments using music and speech signals [27]. It has been concluded that the perceptual effects of nonlinear distortion are generally greater than those of linear distortion, except when the linear distortion is severe. Similarly, for speech quality, linear distortion has been found to be generally less objectionable than nonlinear distortion [23].

1.1.3 Audio Signal Enhancement

The abundance of potential audio signal distortion mechanisms throughout the audio signal path and their negative effect on the quality and intelligibility of audio signals make it indispensable to design and apply effective *audio signal enhancement* algorithms. The goal of audio signal enhancement algorithms is to improve the quality and/or intelligibility of an audio signal that is degraded by a given linear and/or nonlinear distortion process, by applying some form of real-time digital signal processing.

Most audio signal enhancement algorithms assume a model for the distortion process under consideration. Figure 1.2 shows a generic distortion process acting on a clean audio signal $\mathbf{x}$, which results in a distorted audio signal $\mathbf{y}$. Note that throughout this thesis, audio signals are represented using vectors containing the audio signal samples as their elements. The distortion process is typically modeled by a linear or nonlinear distortion model $\mathbf{f}(\mathbf{x}, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ are the distortion model parameters. As the properties of the distortion process can change over time, the model parameters $\boldsymbol{\theta}$ can be time-varying. Notable examples are the change of reverberation parameters due to a change in the room acoustics [32], and the change of loudspeaker parameters due to temperature changes and ageing [33].

Audio signal enhancement algorithms can be classified into two types, depending on whether they are applied to the audio signal *before* or *after* the distortion process. The former algorithms are called *precompensation algorithms*, the latter algorithms are called *recovery algorithms*. Precompensation algorithms are typically applied in situations where the clean audio signal $\mathbf{x}$ can be observed and altered prior to the distortion process, e.g. prior to reproduction through a


$\hat{\boldsymbol{\theta}}$
$\mathbf{x}$
Precompensation algorithm
$\mathbf{v}$
Distortion process
$\mathbf{f}(\mathbf{v}, \boldsymbol{\theta})$
$\mathbf{y}^*$
Parameter estimation
$\hat{\boldsymbol{\theta}}$
$\mathbf{x}$
Distortion process
$\mathbf{f}(\mathbf{x}, \boldsymbol{\theta})$
$\mathbf{y}$
Recovery algorithm
$\mathbf{y}^*$

(a) Precompensation algorithms

(b) Recovery algorithms

Figure 1.3: Types of audio signal enhancement algorithms.

distorting loudspeaker. Recovery algorithms are typically applied in situations where the clean audio signal $\mathbf{x}$ cannot be observed nor altered prior to the distortion process, but the distorted audio signal $\mathbf{y}$ can be observed and altered after the distortion process, e.g. after reverberation distortion has been added to the audio signal.

The operation of a generic precompensation algorithm is illustrated in Figure 1.3(a). It is seen that the precompensation algorithm is applied before the distortion process acts onto the audio signal. The precompensated audio signal $\mathbf{v}$ is computed based on the clean audio signal $\mathbf{x}$ and the estimated distortion model parameters $\hat{\boldsymbol{\theta}}$. The enhanced audio signal $\mathbf{y}^*$ is the result of applying the precompensated audio signal $\mathbf{v}$ to the distortion process. In this set-up, it is necessary to estimate the distortion model parameters $\hat{\boldsymbol{\theta}}$ during a separate off-line estimation procedure, as it will be assumed that it is not possible to feed back the enhanced audio signal $\mathbf{y}^*$ on-line.

The operation of a generic recovery algorithm is shown in Figure 1.3(b). It is seen that the recovery algorithm is applied after the distortion process acts onto the audio signal. The enhanced audio signal $\mathbf{y}^*$ is computed based on the distorted audio signal $\mathbf{y}$ and the estimated distortion model parameters $\hat{\boldsymbol{\theta}}$. In this set-up, it is necessary to estimate the distortion model parameters $\hat{\boldsymbol{\theta}}$ during an estimation procedure, which will be assumed to be performed on-line and blindly.

We can define two crucial requirements for any on-line audio signal enhancement algorithm:

1. The algorithm should consistently improve a desired perceptual attribute

(audio quality, speech quality, speech intelligibility), i.e. the perceptual attribute should be better for the enhanced audio signal $\mathbf{y}^*$ compared to the distorted audio signal $\mathbf{y}$. Ideally, the enhanced audio signal $\mathbf{y}^*$ is equal to the clean audio signal $\mathbf{x}$.

2. The algorithm should be able to run under strict constraints regarding computation time, resource usage and power consumption, as will be typically imposed by (mobile) audio devices.

In the next sections, we will discuss the state of the art and the prevailing challenges for precompensation algorithms (see Section 1.2) and recovery algorithms (see Section 1.3). For both types of audio signal enhancement algorithms, the analysis will focus on two commonly encountered yet challenging audio signal distortion processes. In Section 1.4, we will outline a novel audio signal enhancement framework for overcoming the limitations of existing audio signal enhancement algorithms, which is based on the application of *embedded optimization* and *perceptual models.*

1.2 Precompensation Algorithms

From Figure 1.3(a), we can define the following steps in the operation of a generic precompensation algorithm:

1. Off-line selection of a suitable distortion model $\mathbf{f}(\mathbf{v}, \boldsymbol{\theta})$.
2. Off-line estimation of distortion model parameters $\hat{\boldsymbol{\theta}}$.
3. On-line computation of precompensated audio signal $\mathbf{v}$.

We will now review the problem statement and state of the art of precompensation algorithms for mitigating *hard clipping distortion* (subsection 1.2.1) and *loudspeaker distortion* (subsection 1.2.2), thereby focusing on the efficiency and limitations in performing the three steps mentioned above.

1.2.1 Hard Clipping Precompensation

Hard clipping is a nonlinear distortion process commonly encountered in audio applications, and can occur during the recording, mastering, storage, transmission and reproduction stages of the audio signal path. When hard clipping occurs, the amplitude of the clean audio signal is cut off such that no sample amplitude exceeds a given amplitude range $[L, U]$. This introduces different kinds of unwanted nonlinear distortion into the audio signal such as odd harmonic distortion, intermodulation distortion and aliasing distortion [34]. In a series of listening experiments performed on normal hearing listeners [2] and hearing-impaired listeners [35], it was concluded that the application of hard clipping to audio signals has a significant negative effect on perceived audio quality scores, irrespective of the subject's hearing acuity. Moreover, it was concluded that hard clipping distortion reduces speech intelligibility, both for normal-hearing and hearing-impaired listeners [31]. Hard clipping precompensation algorithms typically focus on reducing the negative effects of hard clipping on the resulting audio quality. The operation of a generic hard clipping precompensation algorithm is shown in Figure 1.4.

Distortion Model Selection

The selection of a suitable distortion model is straightforward in this case. As shown in Figure 1.4, the hard clipping distortion can be exactly modeled using a memoryless hard clipping nonlinearity that is linear in the amplitude range $[L, U]$, and abruptly saturates when exceeding this amplitude range.

Distortion Model Parameter Estimation

The parameters of the distortion model are the lower clipping level $L < 0$ and the upper clipping level $U > 0$ of the memoryless nonlinearity. A common approach to estimate L and U is to detect the occurrence of hard clipping based on the distorted audio signal. Such non-intrusive hard clipping detection methods rely on the inspection of anomalities in the amplitude histogram [36] in order to detect the occurrence of hard clipping and estimate the associated parameters L and U. These methods are very accurate if the detection works on the raw hard clipped audio signal, but are less accurate when the hard clipped audio signal was perceptually encoded prior to detection, in which case robust detection methods are necessary [37].

Precompensation Operation

Hard clipping precompensation algorithms aim to preventively limit the digital audio signal with respect to the estimated allowable amplitude range $[\hat{L},\hat{U}]$ of the subsequent hard clipping distortion process. Ideally, the precompensated audio signal $\mathbf{v}$ can then pass through the hard clipping distortion process without being altered, i.e. $\mathbf{y}^* = \mathbf{v}$. The precompensation algorithm is obviously expected to add minimal distortion to the clean audio signal $\mathbf{x}$. We can classify existing hard clipping precompensation algorithms into *limiting* algorithms and *soft clipping* algorithms.

Limiting algorithms (or *limiters*) aim to provide control over the amplitude peaks exceeding $[\hat{L},\hat{U}]$ in the clean audio signal $\mathbf{x}$, while changing the dynamics and frequency content of the audio signal as little as possible [1]. Limiters are essentially amplifiers with a time-varying gain that is automatically controlled by the measured peak level of the clean audio signal $\mathbf{x}$. The attack time and release time parameters specify how fast the gain is changed according to mea-

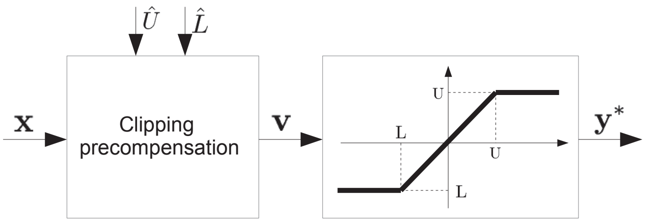


Figure 1.4: Hard clipping precompensation.

sured peaks in the clean audio signal $\mathbf{x}$. The attack time parameter defines how fast the gain is decreased when the input signal level rapidly increases, while the release time parameter defines how fast the gain is restored to its original value when the input signal level rapidly decreases [38]. The setting of these parameters entails a trade-off between distortion avoidance and peak limiting performance, as the gain should be as smooth as possible for not having audible artefacts, yet at the same time it should vary fast enough to suppress signal peaks [39][40].

Soft clipping algorithms instantaneously limit the clean audio signal $\mathbf{x}$ to the estimated allowable amplitude range $[\hat{L},\hat{U}]$ by applying a soft memoryless nonlinearity, i.e. one having a gradual transition from the linear zone to the nonlinear zone. In fact, soft clipping algorithms are related to limiting algorithms in that they can be viewed as limiters having an infinitely small attack and release time [1]. In general, soft memoryless nonlinearities introduce less perceptible artefacts as compared to hard memoryless nonlinearities, because of the lower level of the introduced harmonic distortion and aliasing distortion [41]. Numerous soft memoryless nonlinearities have been proposed, such as hyperbolic tangent, inverse square root, parabolic sigmoid, cubic sigmoid, sinusoidal, and exponential soft memoryless nonlinearities [42][43].

While both limiting algorithms and soft clipping algorithms have been shown to work fairly well for mitigating the effects of specific hard clipping distortion processes, several limitations of these approaches can be indicated. Firstly, these algorithms are governed by a set of tunable parameters, such as the attack time and release time for limiting approaches, and the shape parameters of the applied soft memoryless nonlinearity in soft clipping approaches. The relation between the parameter settings and the resulting enhancement of the desired perceptual attribute is generally unclear, leading in many cases to an ad hoc and trial-and-error based parameter tuning procedure. Secondly, as these approaches act directly on the amplitude of the clean time-domain audio signal, it is difficult to adapt to time-varying frequency characteristics of the clean audio signal. Lastly, as the properties of human sound perception are not incorporated into these approaches, it is not possible to focus on enhancing a

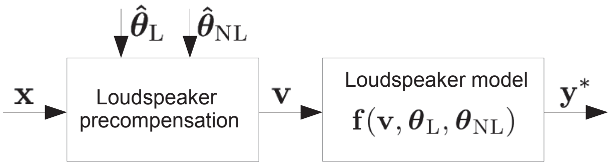

$\hat{\boldsymbol{\theta}}_{\mathrm{L}}$
$\hat{\boldsymbol{\theta}}_{\mathrm{NL}}$
$\mathbf{x}$
Loudspeaker precompensation
$\mathbf{v}$
Loudspeaker model
$\mathbf{f}(\mathbf{v}, \boldsymbol{\theta}_{\mathrm{L}}, \boldsymbol{\theta}_{\mathrm{NL}})$
$\mathbf{y}^*$

Figure 1.5: Loudspeaker precompensation.

given perceptual attribute of the audio signal.

1.2.2 Loudspeaker Precompensation

Loudspeaker distortion is a form of combined linear and nonlinear distortion incurred when an audio signal is reproduced through a loudspeaker system having a non-ideal response. At low amplitudes, the loudspeaker behaviour is almost linear and nonlinear signal distortion is negligible. However, at higher amplitudes nonlinear distortion occurs, and is notably prominent in small and low-cost loudspeakers, which are ubiquitous in mobile devices [44]. Linear loudspeaker distortion is typically perceived as affecting timbre or tone quality, whereas nonlinear loudspeaker distortion is typically perceived as adding harshness or noisiness, or as the perception of crackles or clicks. The presence of linear and nonlinear loudspeaker distortion has been found to result in a significant degradation of the perceived audio quality, both when present separately [2] and simultaneously [27]. Loudspeaker precompensation algorithms typically focus on reducing the negative effects of loudspeaker distortion on the resulting audio quality.

Distortion Model Selection

The selection of a suitable model accurately representing the linear and nonlinear loudspeaker distortion is not a trivial task. Loudspeaker models can be classified as *linear* loudspeaker models or *nonlinear* loudspeaker models. Knowledge of the physical nonlinear mechanisms inside the loudspeaker can be incorporated to different degrees, leading to a further subclassification in *white-box*, *grey-box* or *black-box* nonlinear loudspeaker models [45].

Traditionally, loudspeakers have been modeled using linear systems, such as FIR filters [46] and IIR filters [47]. Warped FIR and IIR filters [48], as well as Kautz filters [49] have been proposed in order to allow for a better frequency resolution allocation, radically reducing the required filter order.

Nonlinear loudspeaker behaviour can be taken into consideration by using nonlinear loudspeaker models. The most widely used white-box nonlinear loudspeaker models are physical low-frequency lumped parameter models, which take into account nonlinearities in the motor part and the mechanical part of the loudspeaker [50]. Given the relative complexity of such physical loudspeaker models and their limitation to low frequencies and low-order nonlinearities, simpler and more efficient grey-box nonlinear loudspeaker models have been proposed, such as Hammerstein models [51], cascades of Hammerstein models [52], and Wiener models [53]. These models are composed of a linear dynamic part and a nonlinear static part, capable of incorporating prior information on the linear and nonlinear distortion mechanisms in the loudspeaker. Black-box models have also been applied to loudspeaker modeling, e.g. time-domain NARMAX models [54], or frequency-domain Volterra models [55]. A major drawback of Volterra models is that the number of parameters grows exponentially with the model order, in contrast to Hammerstein and Wiener models.

Distortion Model Parameter Estimation

As shown in Figure 1.5, the loudspeaker model parameters can in general be divided into a set of model parameters $\boldsymbol{\theta}_{\mathrm{L}}$ related to the linear part of the model and a set of model parameters $\boldsymbol{\theta}_{\mathrm{NL}}$ related to the nonlinear part of the model. For linear loudspeaker models, only the parameter set $\boldsymbol{\theta}_{\mathrm{L}}$ has to be estimated. For nonlinear loudspeaker models, both the parameter sets $\boldsymbol{\theta}_{\mathrm{L}}$ and $\boldsymbol{\theta}_{\mathrm{NL}}$ have to be estimated.

The parameters of linear loudspeaker models, grey-box and black-box nonlinear loudspeaker models are mostly estimated by exciting the loudspeaker with audio-like signals, e.g. random phase multisines [56], and recording the reproduced signal. The parameters of white-box low-frequency lumped parameter models can be estimated by exciting the loudspeaker with an audio-like signal and measuring the voice coil current [15], or the voice coil displacement using an optical sensor [57].

While the parameter estimation of linear loudspeaker models can be performed using standard linear identification methods, the parameter estimation of nonlinear loudspeaker models is a challenging problem. Hammerstein model parameter estimation requires the solution of a bi-convex optimization problem, having an objective function featuring cross products between parameters in $\boldsymbol{\theta}_{\mathrm{L}}$ and parameters in $\boldsymbol{\theta}_{\mathrm{NL}}$. Techniques[2] to solve this bi-convex optimization problem include the iterative approach [59], the overparametrization approach [60], and the subspace approach [61]. Wiener model parameter estimation methods have been derived along the lines of their Hammerstein counterparts, resulting in the same categories of approaches for solving the bi-convex optimization problem [62]. Volterra model parameters can be estimated using adaptive algorithms such as NLMS [55].

[2] A nice overview of different Hammerstein model identification methods is given in [58].

Precompensation Operation

The operation of a generic loudspeaker precompensation algorithm is shown in Figure 1.5. The idea is to reduce the linear and nonlinear distortion effects caused by the loudspeaker, by applying a precompensation step to the clean audio signal $\mathbf{x}$ before feeding it to the loudspeaker input. The estimated loudspeaker model parameters $\hat{\boldsymbol{\theta}}_{\mathrm{L}}$ and $\hat{\boldsymbol{\theta}}_{\mathrm{NL}}$ are used in the precompensation.

When using linear loudspeaker models, precompensation consists in performing *linear equalization* of the loudspeaker by computing (based on $\hat{\boldsymbol{\theta}}_{\mathrm{L}}$) and applying an inverse digital filter to the audio signal. An ideal linear equalization would result in a reproduction channel having a flat frequency response and a constant group delay. Among the proposed equalization approaches, we mention the distinction between direct inversion and indirect inversion approaches, and between minimum-phase and nonminimum-phase designs. In general, the performance of these equalization approaches is seen to largely depend on the stationarity and the accuracy of the loudspeaker models [49].

When using nonlinear loudspeaker models, precompensation consists in performing either *linearization* or *full equalization* of the loudspeaker. The aim of linearization is to make the reproduction channel a linear system, thereby compensating for the nonlinear distortion in the loudspeaker [63]. The aim of full equalization is to make the reproduction channel transparent, thereby compensating for both the linear and nonlinear distortion in the loudspeaker.

Nonlinear loudspeaker precompensation methods for performing linearization have been proposed for white-box, grey-box and black-box loudspeaker models. For white-box low-frequency lumped parameter models, seminal linearization methods are based on the application of nonlinear inversion [50] and a mirror filter [64]. A control-theoretic feedback linearization approach was theoretically shown to allow for exact linearization under certain assumptions [65], and this approach was modified to achieve a satisfactory approximate linearization in practice [66]. For grey-box Wiener and Hammerstein loudspeaker models, linearization methods have been proposed based on the coherence criterion [51] and polynomial root finding [67]. For black-box Volterra loudspeaker models, a p-th order inverse model was succesfully applied to achieve loudspeaker linearization [68]. The main disadvantage of these methods resides in their high computational complexity.

Nonlinear loudspeaker precompensation methods for performing full equalization rely on the computation of an inverse nonlinear loudspeaker model. However, the exact inverse of the nonlinear loudspeaker model only exists in specific cases. For Hammerstein and Wiener loudspeaker models, an exact inverse only exists if the inverse of the static nonlinearity exists. Volterra loudspeaker models in general do not allow for computing an exact inverse model. As a consequence, practical full equalization methods rely on the computation of an inexact inverse model, which brings along problems with both the stability and the computational complexity of these methods [44].

In conlusion, several general limitations of the existing approaches for loudspeaker precompensation can be indicated. Firstly, their fairly high computational complexity conflicts with the requirement to perform loudspeaker compensation in real time on mobile audio devices. Secondly, as the properties of human sound perception are not incorporated into these approaches, it is not possible to focus on enhancing a given perceptual attribute of the audio signal.

1.3 Recovery Algorithms

From Figure 1.3(b), we can define the following steps in the operation of a generic recovery algorithm:

1. Off-line selection of a suitable distortion model $\mathbf{f}(\mathbf{x}, \boldsymbol{\theta})$.
2. On-line blind estimation of distortion model parameters $\hat{\boldsymbol{\theta}}$.
3. On-line computation of the enhanced audio signal $\mathbf{y}^*$.

We will now review the problem statement and state-of-the-art of recovery algorithms for enhancing audio signals degraded by *hard clipping distortion* (subsection 1.3.1) and *reverberation distortion* (subsection 1.3.2), thereby focusing on the efficiency and limitations in performing the three steps mentioned above.

1.3.1 Declipping

In subsection 1.2.1, it was shown that hard clipping is a nonlinear distortion process that can occur in almost any stage of the audio signal path, and has a significant negative effect on the audio quality, speech quality and speech intelligiblity. In situations where hard clipping can not be anticipated for, one has to perform declipping, i.e. the recovery of the clean audio signal $\mathbf{x}$ based on the hard clipped audio signal $\mathbf{y}$. The operation of a generic declipping algorithm is shown in Figure 1.6.

Distortion Model Selection and Parameter Estimation

As mentioned in subsection 1.2.1, the selection of a suitable distortion model is straightforward, i.e. the hard clipping distortion can be exactly modeled using a memoryless hard clipping nonlinearity that is linear in the amplitude range $[L, U]$, and abruptly saturates when exceeding this amplitude range. The parameters of the distortion model are the lower clipping level $L < 0$ and the upper clipping level $U > 0$ of the memoryless nonlinearity. These parameters can be estimated based on the hard clipped audio signal $\mathbf{y}$, using histogram methods.

Recovery Operation

Several approaches to the declipping problem have been proposed. A first approach is based on performing an *interpolation* procedure to recover the clipped signal samples based on the knowledge of the unclipped signal samples. Interpolation algorithms differ in particular in the a priori knowledge and assumptions on the clean audio signal $\mathbf{x}$ that are incorporated into the interpolation procedure. Autoregressive [69], sinusoidal [70] and statistical audio signal models [71] have been used, as well as restrictions on the spectral envelope [72], bandwidth [73], and time-domain amplitude [71][73][74] of the clean audio signal. A second approach tackles the declipping problem as a *supervised learning* problem, in which the temporal and spectral properties of clean and clipped audio signals are learned through an artificial neural network [75], or a Hidden Markov Model (HMM) [76].

The third and most recent approach addresses the declipping problem in the framework of *compressed sensing* (CS). In the CS framework, declipping is formulated and solved as a sparse signal recovery problem, where one takes advantage of the sparsity of the clean audio signal (in some basis or dictionary) in order to recover it from a subset of its samples. Sparse signal recovery methods for declipping differ in the sparsifying basis or dictionary that is used to represent the clean audio signal, and in the optimization procedure that is used for computing the recovered audio signal. Commonly used sparse audio signal representations include the Discrete Fourier Transform (DFT) basis [77], the overcomplete Discrete Fourier Transform (DCT) dictionary [78][79], and the overcomplete Gabor dictionary [80]. In order to solve the sparse signal recovery optimization problem, existing algorithms such as Orthogonal Matching Pursuit (OMP) [78], Iterative Hard Thresholding (IHT) [79], Trivial Pursuit (TP) [77] and reweighted L1-minimization [77] have been adapted in order to incorporate constraints specific to the declipping problem. For some of these sparse signal recovery methods for declipping, deterministic recovery guarantees have been derived in [81][82].

In conclusion, we can point out a general limitation of the existing declipping methods. Whereas all these methods do include a model of the clean audio signal, they do not incorporate a model of the human sound perception, making it impossible to focus on enhancing a given perceptual attribute of the audio signal.

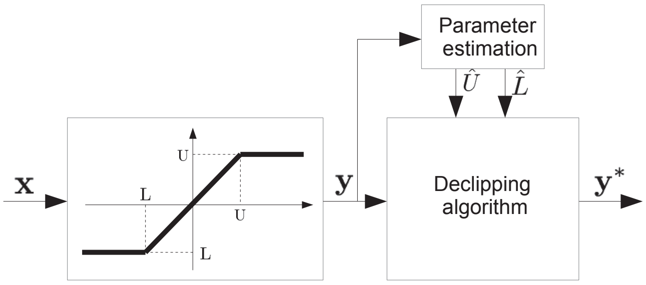


Figure 1.6: Declipping.

1.3.2 Dereverberation

Reverberation is a form of linear distortion introduced by the acoustics of the room in which the clean audio signal is produced and recorded. When the clean audio signal is produced in a closed acoustic environment, it is partially reflected by the walls, the floor, and ceiling of the room. As a result, not only the clean audio signal is picked up by the recording device, but also several delayed and attenuated replicas of the clean audio signal [3]. Linear distortion caused by reverberation is known to add spaciousness and coloration to the sound. For music signals, this not necessarily an undesired property. However, for speech signals, reverberation makes the talker sound far away from the microphone and is known to have a significant negative impact on both speech quality and speech intelligibility [28]. Speech intelligibility is especially degraded by reverberation for non-native listeners and for listeners with hearing impairments [29]. Dereverberation algorithms typically focus on reducing the negative effects of reverberation on the resulting speech intelligibility.

Distortion Model Selection

The most simple and popular room acoustics models are time-domain FIR and IIR filters. The necessary filter order mainly depends on the *reverberation time* of the room, which is defined as the time taken for the reverberant energy to decay by 60 dB once a stationary sound source has been shut off. The reverberation time for a room is governed by the room geometry and the reflectivity of the room surfaces and objects within the room. As the reverberation time in typical rooms can be expected to be within the range 0.1 to 1 seconds, FIR filters of several thousand taps are typically necessary for commonly used sampling frequencies [28]. Numerical room acoustics models include Finite Element methods, Boundary Element methods [83], and Finite Difference Time-Domain (FDTD) methods [84], which all consider numerical discretizations of the wave equation in situations where no analytical solution can be computed. For these

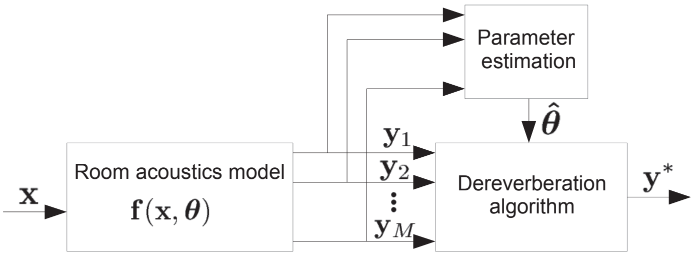


Figure 1.7: Dereverberation.

methods, the incorporation of appropriate boundary conditions and the geometrical description of objects within the acoustic environment is highly challenging. Ray-tracing models represent sound waves by rays that can be traced from sound source to observer, assuming all the reflections are specular [85]. These modeling methods are typically computationally intensive.

Distortion Model Parameter Estimation and Recovery Operation

As illustrated in Figure 1.7, a generic *dereverberation* algorithm aims to remove reverberation from $M \geq 1$ available microphone signals recorded in a reverberant environment. Dereverberation is generally still considered one of the most challenging problems in the area of acoustic signal enhancement [28]. One of the major difficulties is that dereverberation is an inverse problem, i.e., one aims at inverting the behaviour of the room. Furthermore, dereverberation is a blind problem, in which both the clean audio signal $\mathbf{x}$ and the room acoustics model parameters $\boldsymbol{\theta}$ are unknown. The state of the art in dereverberation can be classified into three categories: (1) beamforming using microphone arrays, (2) speech enhancement approaches to dereverberation, and (3) blind system identification and inversion.

In the beamforming approach to dereverberation, the microphone signals obtained in a microphone array are delayed and weighted or filtered such that an appropriate spatial beam pattern is created. The beam pattern should be such that a high array sensitivity is obtained in the direction of the sound source. The design of fixed and adaptive beamformers is a mature research area when considering the problem of removing uncorrelated noise from multi-microphone recordings [86]. However, dereverberation requires the removal of undesired room reflections that are correlated with the dry signal. This is feasible either by employing multiple beamformers for steering additional beams towards the early reflections [87], or by designing a matched filter beamformer using the time-reversed RIRs from the sound source to the array microphones [88]. In both cases, a room acoustics model is required.

Several speech enhancement approaches to dereverberation have been explored. An early approach was shown to be capable of compensating for early reflections in the cepstral domain [89]. A more successful technique is based on linear prediction [90], and exploits the observation that reverberation only alters the linear prediction (LP) residual of a speech signal, but not the autoregressive (AR) coefficients. Dereverberated speech can then be synthesized from the AR coefficients, using an enhanced LP residual signal obtained through matched filtering [91], kurtosis maximization [92], or spatiotemporal averaging [93]. Another speech enhancement approach to dereverberation is spectral subtraction, which can be applied in a single-channel [94] or multi-channel [95] scenario. Again, a room acoustics model is necessary for being able to estimate and subtract the spectral contribution of reverberation from the microphone signals.

A promising yet challenging approach to dereverberation is blind system identification and inversion. The aim is to identify the room acoustics model from the observed microphone signals, to invert this model, and to process the microphone signals with the inverse model to obtain an estimate of the clean audio signal. In a multi-channel scenario, the blind system identification problem can be elegantly formulated as an eigenvalue problem, for which least-squares [96], subspace [97], and adaptive filtering [98] solutions have been proposed. However, the application of these algorithms to blind identification of room acoustics was found to be nontrivial due to nonstationarity, robustness, and computational issues [28]. The inversion of the identified room acoustics model can be performed in a single-channel scenario [99], although multi-channel channel inversion was found to be computationally more promising and more appealing [100]. More recently, subband [101] and adaptive [102] implementations for multi-channel inversion have been proposed, as well as regularized algorithms to improve robustness against model identification errors [32][103][104].

1.4 Embedded Optimization Framework for Perceptual Audio Signal Enhancement

From the overview of the state-of-the-art algorithms for hard clipping precompensation, loudspeaker precompensation, declipping and dereverberation, we can point out two major general limitations of existing audio signal enhancement algorithms:

- The properties of human sound perception are most often not incorporated into traditional audio signal enhancement algorithms. As a result, the relation between the algorithm parameters and the perceptual attribute to enhance is not clearly defined, nor can it be ensured that the desired perceptual attribute will indeed be enhanced.
- Most traditional audio signal enhancement algorithms rely on a sequence

of a (non)linear system identification, the inversion of the identified system, and the application of the inverted system to the audio signal. This sequence is mostly computationally intensive and often contains ill-conditioned steps.

The goal of this thesis is to overcome the limitations of existing audio signal enhancement approaches, by exploring a new audio signal enhancement framework. In this *embedded optimization* framework, the audio signal enhancement problem at hand is reformulated as a per-frame numerical optimization problem, aimed at enhancing a desired perceptual attribute.

1.4.1 Embedded Optimization

Traditionally, numerical optimization has been applied in speech and audio signal processing applications for tasks such as FIR filter and filterbank design [105], microphone array weight design [106], and feature learning in audio classification [107] or speech recognition [108]. These design tasks are typically performed off-line, and the optimization is carried out on time scales of seconds, minutes or hours, often without strict computation time constraints. Recently, the growing computational power and algorithmic developments have opened up the possibility of embedding numerical optimization directly into real-time signal processing algorithms in order to carry out nonlinear processing on the signal itself. In this *embedded optimization* paradigm, the optimization variables are signal samples instead of filter weights or model parameter values, and the optimization can be performed on-line with strict computation time constraints, on time scales as short as milliseconds or microseconds [109].

This thesis investigates the development of an embedded optimization framework for audio signal enhancement, and its application to both precompensation algorithms and recovery algorithms. In Figure 1.8(a), the concept of an embedded optimization algorithm for the precompensation of audio signals is visualized. The algorithm operates on short-time clean audio signal frames $\mathbf{x}_m$ of N samples, where m denotes the frame index. The precompensated audio signal frame $\mathbf{v}_m$ is computed by solving a per-frame constrained optimization problem of the following form:

$$\underset{\mathbf{v}_m \in \mathbb{R}^N}{\text{minimize}}\ d_m(\mathbf{x}_m, \hat{\mathbf{y}}_m^*) \quad \text{subject to} \quad \hat{\mathbf{y}}_m^* = \mathbf{f}(\mathbf{v}_m, \hat{\boldsymbol{\theta}}_m) \tag{1.1}$$

where a given distance measure d_m between the clean audio signal frame $\mathbf{x}_m$ and the predicted enhanced audio signal frame $\hat{\mathbf{y}}_m^*$ is minimized, and the constraints model the subsequent distortion process using distortion model parameters $\hat{\boldsymbol{\theta}}_m$ derived from an off-line estimation procedure. In Figure 1.8(b), the concept of an embedded optimization algorithm for the recovery of audio signals is visualized. The enhanced audio signal frame $\mathbf{y}_m^*$ is computed by solving a per-frame constrained optimization problem of the following form:

$$\underset{\mathbf{y}_m^* \in \mathbb{R}^N}{\text{minimize}} \; d_m(\mathbf{y}_m^*) \quad \text{subject to} \quad \mathbf{y}_m = \mathbf{f}(\mathbf{y}_m^*, \hat{\boldsymbol{\theta}}_m) \tag{1.2}$$

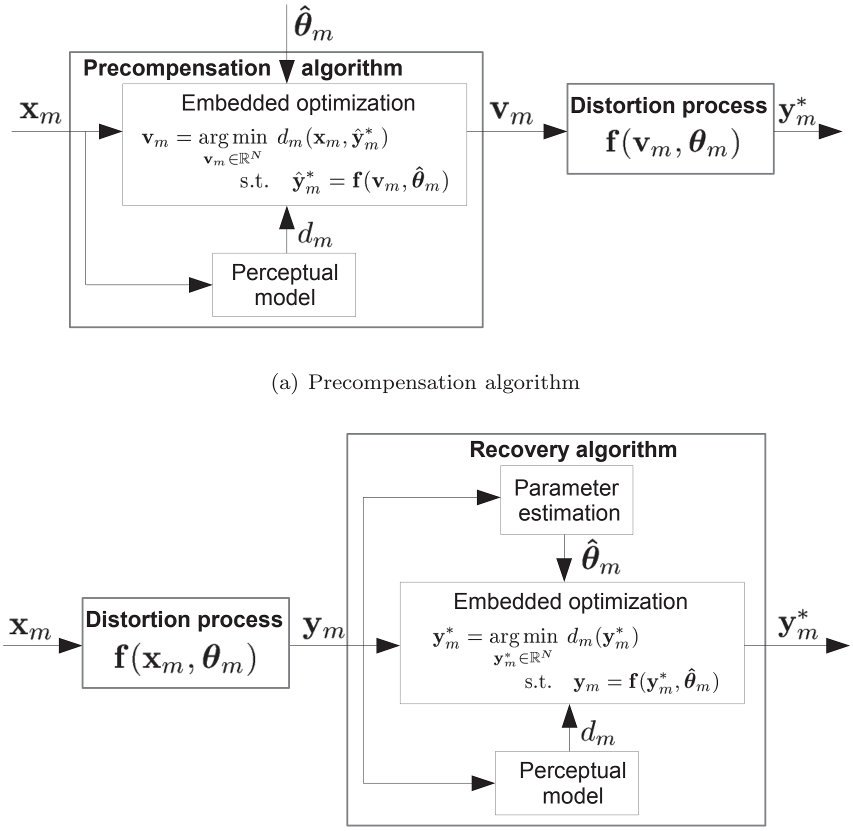
where a given measure d_m of the enhanced audio signal frame $\mathbf{y}_m^*$ is minimized, and the constraints model the preceding distortion process using distortion model parameters $\hat{\boldsymbol{\theta}}_m$ derived from an on-line estimation procedure.

The embedded optimization framework allows us to compute in real time an enhanced audio signal frame $\mathbf{y}_m^*$ that is optimal according to any desired mathematical measure d_m. For audio signal enchancement applications, the most relevant measure to minimize is the perceived degradation of a selected perceptual attribute (e.g. audio quality, speech quality, speech intelligibility) between the clean audio signal frame $\mathbf{x}_m$ and the enhanced signal frame $\mathbf{y}_m^*$. The translation from the *subjective* perceptual attribute degradation to the *objective* mathematical measure d_m can be performed using a suitable model of human sound perception, as will be discussed next.

1.4.2 Perceptual Models

The human perception of sounds is a complex process involving both auditory and cognitive mechanisms. In the last decades, considerable research efforts have been spent in order to thoroughly understand and accurately model the essential properties of human sound perception. As giving a complete overview of these findings is largely beyond the scope of this thesis, we will focus on pointing out a few properties of the human perception of sounds which are most relevant to the perception of linear and nonlinear distortion.

Most important, the sensitivity of the human hearing system is dependent on the frequency of the tone. This effect is characterized by the *absolute threshold of hearing*, which is defined as the required Sound Pressure Level (SPL) of a pure tone such that an average listener will just hear the tone in a noiseless environment. The absolute threshold of hearing has been measured experimentally [110] and is depicted as the solid line in Figure 1.9. It is seen that the human hearing system is most sensitive in the middle part of the frequency range, and less sensitive for the lower and upper parts of the frequency range. Whenever two or more tones are simultaneously presented to the human hearing system, the phenomenon of *simultaneous masking* may occur, where the presence of one stronger tone (the masker) renders imperceptible the simultaneous presence of a weaker tone (the maskee) at a nearby frequency [111]. In Figure 1.9, the strong masker tone is seen to push the absolute treshold of hearing upwards for nearby frequencies, resulting in a *masking treshold* (dotted line), i.e. the required SPL of a tone such that an average listener will just hear the tone in the simultaneous presence of the masker. As the two displayed tones at nearby frequencies are situated below the masking threshold, they will be imperceptible in the simultaneous presence of the masker.


$\hat{\boldsymbol{\theta}}_m$
Precompensation algorithm
$\mathbf{x}_m$
Embedded optimization
$\mathbf{v}_m = \arg\min_{\mathbf{v}_m \in \mathbb{R}^N} d_m(\mathbf{x}_m, \hat{\mathbf{y}}^*_m)$
s.t. $\hat{\mathbf{y}}^*_m = \mathbf{f}(\mathbf{v}_m, \hat{\boldsymbol{\theta}}_m)$
d_m
Perceptual model
$\mathbf{v}_m$
Distortion process
$\mathbf{f}(\mathbf{v}_m, \boldsymbol{\theta}_m)$
$\mathbf{y}^*_m$
Recovery algorithm
Parameter estimation
$\hat{\boldsymbol{\theta}}_m$
$\mathbf{x}_m$
Distortion process
$\mathbf{f}(\mathbf{x}_m, \boldsymbol{\theta}_m)$
$\mathbf{y}_m$
Embedded optimization
$\mathbf{y}^*_m = \arg\min_{\mathbf{y}^*_m \in \mathbb{R}^N} d_m(\mathbf{y}^*_m)$
s.t. $\mathbf{y}_m = \mathbf{f}(\mathbf{y}^*_m, \hat{\boldsymbol{\theta}}_m)$
d_m
Perceptual model
$\mathbf{y}^*_m$

(a) Precompensation algorithm

(b) Recovery algorithm

Figure 1.8: Embedded optimization algorithms for perceptual audio signal enhancement.

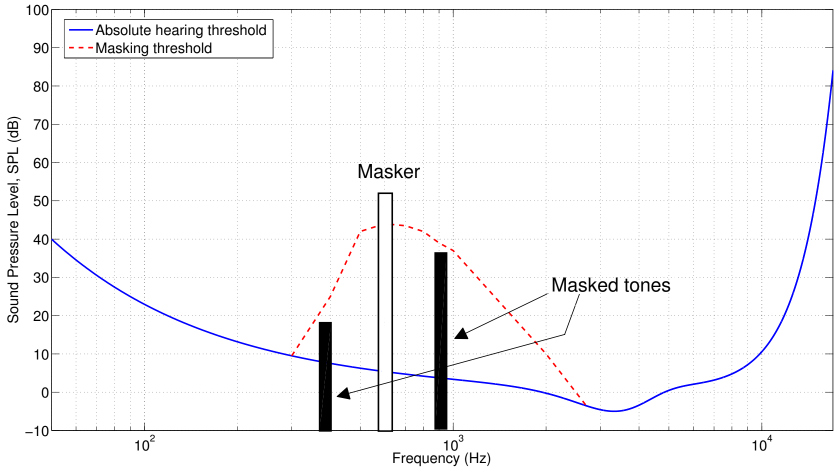


Figure 1.9: Absolute threshold of hearing and masking threshold (adapted from [112]).

Since the absolute threshold of hearing and the masking threshold capture the signal-dependent and frequency-dependent audibility of tones, they form an indispensable component in many *perceptual models*. These perceptual models are typically used to predict the perceived degradation of a given perceptual attribute after the introduction of some form of linear and nonlinear distortion. In this context, they are mostly applied for *off-line* objective *evaluation* purposes, i.e. as an efficient replacement of time-consuming subjective evaluation methods such as formal listening tests. This has led to the development of several objective measures of audio quality [113][114][115], speech quality [116][117] and speech intelligibility [118][119], some of which are standardized. Perceptual models have also been applied for *on-line* algorithm *design* purposes, e.g. in speech [120] and audio signal coding [111], audio signal modeling [121], audio signal requantization [122], audio watermarking [123], and speech enhancement [124].

This thesis investigates how perceptual models can be applied on-line in audio signal enhancement algorithms. The specific goal is to translate the subjective perceptual attribute degradation that we wish to minimize into an adequate objective mathematical measure d_m to be used in the embedded optimization algorithm, as shown in Figures 1.8(a) and 1.8(b).

1.4.3 Main Research Objectives

The main research objectives of this thesis can be stated as follows:

1. Development of an *embedded optimization framework* for audio signal enhancement, resulting in improved audio signal enhancement algorithms for the precompensation and recovery of audio signals subject to different types of linear and nonlinear audio signal distortion. Specific algorithms for hard clipping precompensation, loudspeaker precompensation, declipping and dereverberation are targeted.

2. Incorporation of *perceptual models* in the embedded optimization framework, in order to allow for the consistent enhancement of desired perceptual attributes. The targeted perceptual attribute in this thesis is perceived audio quality.

3. Design of application-tailored *optimization methods* and their efficient *hardware implementation* in order to solve the per-frame optimization problems under strict constraints regarding computation time, resource usage and power consumption.

4. Comparative objective and subjective *evaluation* between the developed embedded optimization algorithms and state-of-the-art audio signal enhancement algorithms.

.

1.5 Thesis Outline and Publications

1.5.1 Chapter-By-Chapter Outline and Contributions

Chapter 2 tackles the problem of *hard clipping precompensation* using an embedded optimization approach, aimed at minimizing the perceptible *nonlinear* audio signal distortion. First, the subjective notion of perceptible nonlinear distortion is translated into a mathematical measure by including frequency weights based on the instantaneous masking threshold, which is computed using a *perceptual model*. Then, hard clipping precompensation is formulated as a sequence of per-frame constrained *convex quadratic optimization problems*. Next, three different structure-exploiting convex optimization methods are proposed for solving the per-frame optimization problems in a fast and reliable way, and their corresponding theoretical complexity bounds are derived. The fastest of these methods is an *optimal projected gradient method*, which is shown to run in real time on a standard PC. Finally, a comparative *objective* evaluation reports significantly higher objective audio quality scores using the embedded optimization algorithm over standard hard clipping precompensation techniques, and this for moderate to high levels of clipping.

Chapter 3 focuses on the problem of *loudspeaker precompensation*, in which the aim is to minimize the combined perceptible *linear* and *nonlinear* audio signal distortion incurred in the loudspeaker, again by using an embedded optimization approach. The loudspeaker is modeled using a grey-box Hammerstein loudspeaker model, i.e. a cascade of a memoryless nonlinearity and a linear FIR filter. First, the subjective notion of perceptible linear and nonlinear distortion is translated into a mathematical measure by including frequency weights based on the instantaneous masking threshold in a similar fashion as in **Chapter 2**. Then, loudspeaker precompensation is formulated as a sequence of per-frame *nonconvex optimization problems*. Next, the resulting per-frame optimization problems are solved efficiently using first-order optimization methods. Depending on the *invertibility* and the *smoothness* of the memoryless nonlinearity in the Hammerstein loudspeaker model, different first-order optimization methods are proposed and their convergence properties are analyzed. Finally, *objective* evaluation experiments using synthetic and identified loudspeaker models show that the embedded optimization loudspeaker precompensation algorithm provides a significant audio quality improvement, especially so at high loudspeaker playback levels.

Chapter 4 reports the set-up and results of *subjective* evaluation experiments for the hard clipping precompensation algorithms of **Chapter 2** and the loudspeaker precompensation algorithms of **Chapter 3**. The subjective evaluation takes the form of a formal listening test including 19 test subjects, each of which is asked to quantify the perceived audio quality of audio signals processed by the different precompensation algorithms. Statistical hypothesis tests on the obtained subject responses lead to the following conclusions. Firstly, the perceived audio quality of audio signals processed by the embedded optimization clipping precompensation algorithm of **Chapter 2** is significantly superior to that of audio signals processed by an existing clipping precompensation algorithm. Secondly, the perceived audio quality of audio signals processed by the embedded optimization loudspeaker precompensation algorithm of **Chapter 3** before being fed into the Hammerstein loudspeaker model is significantly superior to that of audio signals not processed before being fed into the Hammerstein loudspeaker model, and this for moderate to high amplitude levels. Thirdly, there is no reason to believe that the perceived audio quality of audio signals subject to a certain nonlinear hard clipping distortion would be superior to that of audio signals subject to a combination of the same nonlinear hard clipping distortion and an additional linear distortion. Lastly, there is a significant positive correlation between subjective audio quality scores and objective audio quality scores.

Chapter 5 considers the *hardware implementation* on a *field programmable gate array (FPGA)* of the optimal projected gradient method proposed in

Chapter 2 for solving the per-frame convex optimization problems. A couple of organizational changes are proposed in the optimal projected gradient method in order to allow for an efficient FPGA implementation architecture. The architecture selection for implementing the Fast Fourier Transform (FFT) is seen to entail a trade-off between resource usage and throughput. By performing bit-accurate simulations of the FPGA implementation, it is demonstrated that the choice between fixed-point arithmetic and floating-point arithmetic and the selection of the corresponding bit width has an impact on the resulting audio quality as well as on the resource usage and power consumption of the design. The selected FPGA design, which performs 30 optimal projected gradient iterations using 20 fixed-point fraction bits, has a low latency, a reduced power consumption, a reduced resource usage, and at the same time preserves the full audio quality improvement. This design allows to perform the algorithm in real time on a small and portable device.

Chapter 6 focuses on the problem of *declipping*, i.e. the recovery of audio signals degraded by *nonlinear* hard clipping distortion. An embedded optimization algorithm for declipping is proposed that is jointly based on the theory of *compressed sensing (CS)* and on properties of human auditory perception. Declipping is formulated as a sparse signal recovery problem in the DFT domain, in which the recovery is performed by solving a per-frame ℓ_1-norm-type optimization problem. The optimization problem formulation differs from existing CS-based declipping formulations in the inclusion of frequency weights in the ℓ_1-norm, based on the instantaneous masking threshold. As a result, the declipping algorithm is focused on maximizing the perceived audio quality of the declipped signal instead of the physical signal reconstruction quality. Comparative objective and subjective evaluation experiments reveal a significant audio quality increase for the proposed embedded optimization declipping algorithm compared to existing CS-based declipping algorithms.

Chapter 7 focuses on the problem of *multi-microphone dereverberation*, where the aim is to eliminate *linear* reverberation distortion introduced by the acoustics of the room in which the clean audio signal is recorded. An embedded optimization approach to multi-microphone dereverberation is proposed, allowing for a joint estimation of the room acoustics model parameters and the clean audio signal. The proposed approach differs from existing blind system identification and inversion approaches in that it does not require an explicit system inversion. It is shown how the inherently *nonconvex joint estimation problem* can be smoothed by including regularization terms based on a statistical late reverberation model and a sparsity prior for the clean audio signal spectrum. Depending on the included regularization terms, one of three different sequential optimization algorithms for solving the nonconvex optimization problem is applied. A performance evaluation for an example multi-microphone dereverberation scenario shows promising results, thus motivating future research in this direction.

Chapter 8 summarizes the main conclusions of the research presented in this thesis, and points out suggestions for future research.

1.5.2 Included Publications

The following publications are included in this thesis:

Part II: Precompensation algorithms

Chapter 2: Hard Clipping Precompensation
B. Defraene, T. van Waterschoot, H.J. Ferreau, M. Diehl, and M. Moonen, "Real-time perception-based clipping of audio signals using convex optimization," *IEEE Trans. Audio Speech Language Process.*, vol. 20, no. 10, pp. 2657–2671, Dec. 2012.

Chapter 3: Loudspeaker Precompensation
B. Defraene, T. van Waterschoot, M. Diehl, and M. Moonen, "Embedded-optimization-based loudspeaker compensation using a Hammerstein loudspeaker model," *IEEE Trans. Audio Speech Language Process.*, submitted for publication, Nov. 2013.

Part III: Recovery algorithms

Chapter 6: Declipping Using Perceptual Compressed Sensing
B. Defraene, N. Mansour, S. De Hertogh, T. van Waterschoot, M. Diehl, and M. Moonen, "Declipping of audio signals using perceptual compressed sensing," *IEEE Trans. Audio Speech Language Process.*, vol. 21, no. 12, pp. 2627-2637, Dec. 2013.

Chapter 7: Multi-Microphone Dereverberation
T. van Waterschoot, B. Defraene, M. Diehl, and M. Moonen, "Embedded Optimization Algorithms for Multi-Microphone Dereverberation," in *Proc. of the 21st European Signal Process. Conf. (EUSIPCO '13)*, Marrakech, Morocco, Sept. 2013.

Bibliography

[1] U. Zölzer, X. Amatriain, and J. Wiley, *DAFX: digital audio effects.* Wiley Online Library, 2002, vol. 1.

[2] C.-T. Tan, B. C. Moore, and N. Zacharov, "The effect of nonlinear distortion on the perceived quality of music and speech signals," *Journal of the Audio Engineering Society*, vol. 51, no. 11, pp. 1012–1031, 2003.

[3] T. van Waterschoot, "Design and evaluation of digital signal processing algorithms for acoustic feedback and echo cancellation," Ph.D. dissertation, KU Leuven, Leuven, Belgium, Mar. 2009.

[4] U. Zölzer, *Digital audio signal processing.* Wiley. com, 2008.

[5] J. Eargle, *The Microphone Book: From mono to stereo to surround-a guide to microphone design and application.* Taylor & Francis US, 2004.

[6] B. Katz and R. A. Katz, *Mastering audio: the art and the science.* Taylor & Francis US, 2007.

[7] E. Vickers, "The loudness war: Background, speculation, and recommendations," in *Audio Engineering Society Convention 129.* Audio Engineering Society, 2010.

[8] C.-M. Liu, H.-W. Hsu, and W.-C. Lee, "Compression artifacts in perceptual audio coding," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 16, no. 4, pp. 681–695, 2008.

[9] A. Spanias, T. Painter, and V. Atti, *Audio signal processing and coding.* Wiley. com, 2006.

[10] K. C. Pohlmann, *The compact disc: a handbook of theory and use.* AR Editions, Inc., 1989, vol. 5.

[11] S. Godsill, P. Rayner, and O. Cappé, "Digital audio restoration," *Applications of digital signal processing to audio and acoustics*, pp. 133–194, 2002.

[12] M.-Y. Zhu, M. Zhang, X.-Q. Yu, and W.-G. Wan, "Streaming audio packet loss concealment based on sinusoidal frequency estimation in mdct domain," *Consumer Electronics, IEEE Transactions on*, vol. 56, no. 2, pp. 811–819, 2010.

[13] S. Linkwitz, "Investigation of sound quality differences between monopolar and dipolar woofers in small rooms," in *Audio Engineering Society Convention 105.* Audio Engineering Society, 1998.

[14] J. Borwick, *Loudspeaker and headphone handbook.* Taylor & Francis US, 2001.

[15] W. Klippel, "Diagnosis and remedy of nonlinearities in electrodynamical transducers," in *Audio Engineering Society Convention 109.* Audio Engineering Society, 2000.

[16] ——, "Tutorial: Loudspeaker nonlinearitiescauses, parameters, symptoms," *Journal of the Audio Engineering Society*, vol. 54, no. 10, pp. 907–939, 2006.

[17] D. Kolossa and R. Haeb-Umbach, *Robust speech recognition of uncertain or missing data: theory and applications.* Springer, 2011.

[18] D. Fragoulis, G. Rousopoulos, T. Panagopoulos, C. Alexiou, and C. Papaodysseus, "On the automated recognition of seriously distorted musical recordings," *Signal Processing, IEEE Transactions on*, vol. 49, no. 4, pp. 898–908, 2001.

[19] A. Gabrielsson and H. Sjögren, "Perceived sound quality of sound-reproducing systems," *Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 1019–1033, 1979.

[20] S. Bech and N. Zacharov, *Perceptual audio evaluation-Theory, method and application.* Wiley. com, 2007.

[21] J. Benesty, M. M. Sondhi, and Y. Huang, *Springer handbook of speech processing.* Springer, 2008.

[22] M. Waltermann, A. Raake, and S. Moller, "Quality dimensions of narrowband and wideband speech transmission," *Acta Acustica united with Acustica*, vol. 96, no. 6, pp. 1090–1103, 2010.

[23] N. Côté, *T-Labs Series In Telecommunication Services: Integral and Diagnostic Intrusive Prediction of Speech Quality.* Springer, 2011, vol. 1.

[24] A. Raake, *Speech Quality of VoIP: Assessment and Prediction.* Wiley. com, 2007.

[25] B. C. Moore and C.-T. Tan, "Perceived naturalness of spectrally distorted speech and music," *The Journal of the Acoustical Society of America*, vol. 114, p. 408, 2003.

[26] A. Gabrielsson, B. Hagerman, T. Bech-Kristensen, and G. Lundberg, "Perceived sound quality of reproductions with different frequency responses and sound levels," *The Journal of the Acoustical Society of America*, vol. 88, p. 1359, 1990.

[27] B. C. Moore, C.-T. Tan, N. Zacharov, and V.-V. Mattila, “Measuring and predicting the perceived quality of music and speech subjected to combined linear and nonlinear distortion,” *Journal of the Audio Engineering Society*, vol. 52, no. 12, pp. 1228–1244, 2004.

[28] P. A. Naylor and N. D. Gaubitch, *Speech dereverberation.* Springer, 2010.

[29] E. A. P. Habets, “Single- and multi-microphone speech dereverberation using spectral enhancement,” 2007.

[30] K. H. Arehart, J. M. Kates, M. C. Anderson, and L. O. Harvey Jr, “Effects of noise and distortion on speech quality judgments in normal-hearing and hearing-impaired listeners,” *The Journal of the Acoustical Society of America*, vol. 122, p. 1150, 2007.

[31] J. M. Kates and K. H. Arehart, “Coherence and the speech intelligibility index,” *The Journal of the Acoustical Society of America*, vol. 117, p. 2224, 2005.

[32] T. Hikichi, M. Delcroix, and M. Miyoshi, “Inverse filtering for speech dereverberation less sensitive to noise and room transfer function fluctuations,” *EURASIP Journal on Applied Signal Processing*, vol. 2007, no. 1, pp. 62–62, 2007.

[33] B. R. Pedersen and P. Rubak, “Online identification of linear loudspeakers parameters,” in *Audio Engineering Society Convention 122.* Audio Engineering Society, 2007.

[34] F. Foti, “Aliasing distortion in digital dynamics processing, the cause, effect, and method for measuring it: The story of ’digital grunge!’,” in *Preprints AES 106th Conv.*, Munich, Germany, May 1999, Preprint no. 4971.

[35] C.-T. Tan and B. C. J. Moore, “Perception of nonlinear distortion by hearing-impaired people,” *Int. J. Audiol.*, vol. 47, pp. 246–256, May 2008.

[36] T. Otani, M. Tanaka, O. Yasuji, and I. Shusaku, “Clipping detection device and method,” May 21 2009, uS Patent App. 12/470,233.

[37] J. Eaton and P. A. Naylor, “Detection of clipping in coded speech signals,” in *Proc. 2013 European Signal Processing Conference (EUSIPCO-2013)*, Marrakech, Morocco, Sept. 2013.

[38] L. Lu, “A digital realization of audio dynamic range control,” in *Signal Processing Proceedings, 1998. ICSP’98. 1998 Fourth International Conference on*, vol. 2. IEEE, 1998, pp. 1424–1427.

[39] P. Hämäläinen, "Smoothing of the control signal without clipped output in digital peak limiters," in *International Conference on Digital Audio Effects (DAFx)*, 2002, pp. 195–198.

[40] D. Giannoulis, M. Massberg, and J. D. Reiss, "Digital dynamic range compressor design: a tutorial and analysis," *Journal of the Audio Engineering Society*, vol. 60, no. 6, pp. 399–408, 2012.

[41] S. Barbati and T. Serafini, "A perceptual approach on clipping and saturation," 2002.

[42] J. Timoney and V. Lazzarini, "Saturation non-linearities for virtual analog filters," in *Forum Acusticum 2011*, Aalborg, Denmark, Jun. 2011.

[43] S. Enderby and Z. Baracskai, "Harmonic instability of digital soft clipping algorithms," in *Proc. 15th Int. Conference on Digital Audio Effects (DAFx-12)*, York, UK, Sept. 2012.

[44] K. Lashkari, "A modified Volterra-Wiener-Hammerstein model for loudspeaker precompensation," in *Proc. 39th Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, Oct. 2005, pp. 344–348.

[45] P. Brunet and B. Shafai, "New trends in modeling and identification of loudspeaker with nonlinear distortion," *Bli (t)*, vol. 1000, p. d2x, 2011.

[46] M. Karjalainen, E. Piirilä, A. Järvinen, and J. Huopaniemi, "Comparison of loudspeaker equalization methods based on dsp techniques," *Journal of the Audio Engineering Society*, vol. 47, no. 1/2, pp. 14–31, 1999.

[47] G. Ramos and J. J. Lopez, "Filter design method for loudspeaker equalization based on IIR parametric filters," *Journal of the Audio Engineering Society*, vol. 54, no. 12, pp. 1162–1178, 2006.

[48] M. Karjalainen, E. Piirilä, A. Järvinen, and J. Huopaniemi, "Loudspeaker response equalisation using warped digital filters," in *Proc. of NorSig-96*, 1996, pp. 367–370.

[49] M. Karjalainen and T. Paatero, "Equalization of loudspeaker and room responses using Kautz filters: Direct least squares design," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, pp. 1–13, 2007.

[50] A. J. Kaizer, "Modeling of the nonlinear response of an electrodynamic loudspeaker by a volterra series expansion," *Journal of the Audio Engineering Society*, vol. 35, no. 6, pp. 421–433, 1987.

[51] M. V. K. Shi, G. T. Zhou, "Compensation for nonlinearity in a Hammerstein system using the coherence function with application to nonlinear acoustic echo cancellation," *IEEE Transactions on Signal processing*, vol. 55, no. 12, pp. 5853–5858, 2007.

[52] M. Rébillat, R. Hennequin, É. Corteel, and B. F. Katz, "Identification of cascade of hammerstein models for the description of nonlinearities in vibrating devices," *Journal of Sound and Vibration*, vol. 330, no. 5, pp. 1018–1038, 2011.

[53] P. Gilabert, G. Montoro, and E. Bertran, "On the Wiener and Hammerstein models for power amplifier predistortion," in *Microwave Conference Proceedings, 2005. APMC 2005. Asia-Pacific Conference Proceedings*, vol. 2. IEEE, 2005, pp. 4–pp.

[54] H.-K. Jang and K.-J. Kim, "Identification of loudspeaker nonlinearities using the narmax modeling technique," *Journal of the Audio Engineering Society*, vol. 42, no. 1/2, pp. 50–59, 1994.

[55] M. Zeller and W. Kellermann, "Advances in identification and compensation of nonlinear systems by adaptive volterra models," in *Signals, Systems and Computers (ASILOMAR), 2010 Conference Record of the Forty Fourth Asilomar Conference on.* IEEE, 2010, pp. 1940–1944.

[56] P. Crama and J. Schoukens, "Computing an initial estimate of a wiener-hammerstein system with a random phase multisine excitation," *Instrumentation and Measurement, IEEE Transactions on*, vol. 54, no. 1, pp. 117–122, 2005.

[57] Y.-T. Tsai, Y.-d. Lee, and J. H. Huang, "Precision identification of nonlinear damping parameter for a miniature moving-coil transducer," in *World Academy of Science, Engineering and Technology*, 2013, pp. 945–948.

[58] F. Ding, X. P. Liu, and G. Liu, "Identification methods for hammerstein nonlinear systems," *Digital Signal Processing*, vol. 21, no. 2, pp. 215–238, 2011.

[59] E.-W. Bai and D. Li, "Convergence of the iterative Hammerstein system identification algorithm," *Automatic Control, IEEE Transactions on*, vol. 49, no. 11, pp. 1929–1940, 2004.

[60] E.-W. Bai, "An optimal two-stage identification algorithm for hammerstein–wiener nonlinear systems," *Automatica*, vol. 34, no. 3, pp. 333–338, 1998.

[61] I. Goethals, K. Pelckmans, J. A. Suykens, and B. De Moor, "Subspace identification of Hammerstein systems using least squares support vector machines," *Automatic Control, IEEE Transactions on*, vol. 50, no. 10, pp. 1509–1519, 2005.

[62] I. Goethals, "Subspace identification for linear, hammerstein and hammerstein-wiener systems," *status: published*, 2005.

[63] W. Frank, R. Reger, and U. Appel, “Loudspeaker nonlinearities-analysis and compensation,” in *Signals, Systems and Computers, 1992. 1992 Conference Record of The Twenty-Sixth Asilomar Conference on.* IEEE, 1992, pp. 756–760.

[64] W. Klippel, “The mirror filter-a new basis for reducing nonlinear distortion and equalizing response in woofer systems,” *Journal of the Audio Engineering Society*, vol. 40, no. 9, pp. 675–691, 1992.

[65] J. Suykens, J. Vandewalle, and J. Van Ginderdeuren, “Feedback linearization of nonlinear distortion in electrodynamic loudspeakers,” *Journal of the Audio Engineering Society*, vol. 43, no. 9, pp. 690–694, 1995.

[66] D. Franken, K. Meerkotter, and J. Waßmuth, “Observer-based feedback linearization of dynamic loudspeakers with AC amplifiers,” *Speech and Audio Processing, IEEE Transactions on*, vol. 13, no. 2, pp. 233–242, 2005.

[67] K. Lashkari and A. Puranik, “Exact linearization of wiener and hammerstein systems loudspeaker linearization,” in *Information, Communications and Signal Processing, 2005 Fifth International Conference on.* IEEE, 2005, pp. 917–920.

[68] W. A. Frank, “An efficient approximation to the quadratic volterra filter and its application in real-time loudspeaker linearization,” *Signal Processing*, vol. 45, no. 1, pp. 97–113, 1995.

[69] P. A. Esquef and L. W. Biscainho, “An efficient model-based multirate method for reconstruction of audio signals across long gaps,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 4, pp. 1391–1400, 2006.

[70] I. Kauppinen and K. Roth, “Audio signal extrapolation–theory and applications,” in *Proc. DAFx*, 2002, pp. 105–110.

[71] S. J. Godsill, P. J. Wolfe, and W. N. Fong, “Statistical model-based approaches to audio restoration and analysis,” *J. New Music Res.*, vol. 30, no. 4, pp. 323–338, 2001.

[72] R. C. Maher, “A method for extrapolation of missing digital audio data,” *Journal of the Audio Engineering Society*, vol. 42, no. 5, pp. 350–357, 1994.

[73] J. S. Abel en J. O. Smith, “Restoring a clipped signal,” in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, May 1991, pp. 1745–1748.

[74] T. Olofsson, “Deconvolution and model-based restoration of clipped ultrasonic signals,” *Instrumentation and Measurement, IEEE Transactions on*, vol. 54, no. 3, pp. 1235–1240, 2005.

[75] D. Navakauskas, "Artificial neural networks for the restoration of noise distorted songs and audio records," Ph.D. dissertation, Vilnius Gedimina Technicial University, 1999.

[76] J. Han, G. J. Mysore, and B. Pardo, "Audio imputation using the non-negative hidden markov model," in *Latent Variable Analysis and Signal Separation.* Springer, 2012, pp. 347–355.

[77] A. J. Weinstein and M. B. Wakin, "Recovering a clipped signal in sparseland," *Sampling Theory in Signal and Image Processing*, 2012.

[78] A. Adler, V. Emiya, M. G. Jafari, M. Elad, R. Gribonval, and M. D. Plumbley, "A constrained matching pursuit approach to audio declipping," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on.* IEEE, 2011, pp. 329–332.

[79] S. Kitić, L. Jacques, N. Madhu, M. P. Hopwood, A. Spriet, and C. De Vleeschouwer, "Consistent iterative hard thresholding for signal declipping," *arXiv preprint arXiv:1303.1023*, 2013.

[80] A. Adler, V. Emiya, M. G. Jafari, M. Elad, R. Gribonval, and M. D. Plumbley, "Audio inpainting," *IEEE Transactions on Audio, Speech and Language Processing*, 2011.

[81] C. Studer, P. Kuppinger, G. Pope, and H. Bölcskei, "Recovery of sparsely corrupted signals," *IEEE Transactions on Information Theory*, vol. 58, no. 5, pp. 3115–3130, May 2012.

[82] C. Studer and R. G. Baraniuk, "Stable restoration and separation of approximately sparse signals," *arXiv preprint arXiv:1107.0420*, 2011.

[83] A. Pietrzyk, "Computer modeling of the sound field in small rooms," in *Audio Engineering Society Conference: 15th International Conference: Audio, Acoustics & Small Spaces.* Audio Engineering Society, 1998.

[84] D. Botteldooren, "Finite-difference time-domain simulation of low-frequency room acoustic problems," *The Journal of the Acoustical Society of America*, vol. 98, p. 3302, 1995.

[85] J. Giner, C. Militello, and A. García, "Ascertaining confidence within the ray-tracing method," *The Journal of the Acoustical Society of America*, vol. 106, p. 816, 1999.

[86] D. Ward and M. Brandstein, "Microphone arrays: Signal processing techniques and applications," 2001.

[87] J. L. Flanagan, A. C. Surendran, and E.-E. Jan, "Spatially selective sound capture for speech and audio processing," *Speech Communication*, vol. 13, no. 1, pp. 207–222, 1993.

[88] S. Affes and Y. Grenier, "A signal subspace tracking algorithm for microphone array processing of speech," *Speech and Audio Processing, IEEE Transactions on*, vol. 5, no. 5, pp. 425–437, 1997.

[89] A. v. Oppenheim, R. Schafer, and T. Stockham Jr, "Nonlinear filtering of multiplied and convolved signals," *Audio and Electroacoustics, IEEE Transactions on*, vol. 16, no. 3, pp. 437–466, 1968.

[90] B. Yegnanarayana and P. S. Murthy, "Enhancement of reverberant speech using lp residual signal," *Speech and Audio Processing, IEEE Transactions on*, vol. 8, no. 3, pp. 267–281, 2000.

[91] S. M. Griebel and M. S. Brandstein, "Microphone array speech dereverberation using coarse channel modeling," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, vol. 1. IEEE, 2001, pp. 201–204.

[92] M. Wu and D. Wang, "A two-stage algorithm for one-microphone reverberant speech enhancement," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 3, pp. 774–784, 2006.

[93] N. D. Gaubitch and P. A. Naylor, "Spatiotemporal averagingmethod for enhancement of reverberant speech," in *Digital Signal Processing, 2007 15th International Conference on*. IEEE, 2007, pp. 607–610.

[94] K. Lebart, J.-M. Boucher, and P. Denbigh, "A new method based on spectral subtraction for speech dereverberation," *Acta Acustica united with Acustica*, vol. 87, no. 3, pp. 359–366, 2001.

[95] E. Habets, "Multi-channel speech dereverberation based on a statistical model of late reverberation," in *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*, vol. 4. IEEE, 2005, pp. iv–173.

[96] G. Xu, H. Liu, L. Tong, and T. Kailath, "A least-squares approach to blind channel identification," *Signal Processing, IEEE Transactions on*, vol. 43, no. 12, pp. 2982–2993, 1995.

[97] S. Gannot and M. Moonen, "Subspace methods for multimicrophone speech dereverberation," *EURASIP Journal on Applied Signal Processing*, vol. 2003, pp. 1074–1090, 2003.

[98] Y. Huang and J. Benesty, "A class of frequency-domain adaptive approaches to blind multichannel identification," *Signal Processing, IEEE Transactions on*, vol. 51, no. 1, pp. 11–24, 2003.

[99] J. Mourjopoulos, P. Clarkson, and J. Hammond, "A comparative study of least-squares and homomorphic techniques for the inversion of mixed phase signals," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'82.*, vol. 7. IEEE, 1982, pp. 1858–1861.

[100] M. Miyoshi and Y. Kaneda, “Inverse filtering of room acoustics,” *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 36, no. 2, pp. 145–152, 1988.

[101] H. Yamada, H. Wang, and F. Itakura, “Recovering of broadband reverberant speech signal by sub-band mint method,” in *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on.* IEEE, 1991, pp. 969–972.

[102] P. A. Nelson, F. Orduna-Bustamante, and H. Hamada, “Inverse filter design and equalization zones in multichannel sound reproduction,” *Speech and Audio Processing, IEEE Transactions on*, vol. 3, no. 3, pp. 185–192, 1995.

[103] W. Zhang, N. D. Gaubitch, and P. A. Naylor, “Computationally efficient equalization of room impulse responses robust to system estimation errors,” in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on.* IEEE, 2008, pp. 4025–4028.

[104] A. Mertins, T. Mei, and M. Kallinger, “Room impulse response shortening/reshaping with infinity-and¡ formula formulatype=,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, no. 2, pp. 249–259, 2010.

[105] S.-P. Wu, S. Boyd, and L. Vandenberghe, “Fir filter design via spectral factorization and convex optimization,” in *Applied and Computational Control, Signals, and Circuits.* Springer, 1999, pp. 215–245.

[106] A. B. Gershman, N. D. Sidiropoulos, S. Shahbazpanahi, M. Bengtsson, and B. Ottersten, “Convex optimization-based beamforming,” *Signal Processing Magazine, IEEE*, vol. 27, no. 3, pp. 62–75, 2010.

[107] R. Grosse, R. Raina, H. Kwong, and A. Y. Ng, “Shift-invariance sparse coding for audio classification,” *arXiv preprint arXiv:1206.5241*, 2012.

[108] G. Heigold, D. Rybach, R. Schlüter, and H. Ney, “Investigations on convex optimization using log-linear hmms for digit string recognition.” in *INTERSPEECH*, 2009, pp. 216–219.

[109] J. Mattingley and S. Boyd, “Real-time convex optimization in signal processing,” *Signal Processing Magazine, IEEE*, vol. 27, no. 3, pp. 50–61, 2010.

[110] E. Terhardt, “Calculating virtual pitch,” *Hearing research*, vol. 1, no. 2, pp. 155–182, 1979.

[111] T. Painter and A. Spanias, “Perceptual coding of digital audio,” *Proceedings of the IEEE*, vol. 88, no. 4, pp. 451–515, 2000.

[112] M. K. Mandal, *Multimedia signals and systems.* Springer, 2003, vol. 716.

[113] T. Thiede, W. C. Treurniet, R. Bitto, C. Schmidmer, T. Sporer, J. G. Beerends, and C. Colomes, "Peaq-the itu standard for objective measurement of perceived audio quality," *Journal of the Audio Engineering Society*, vol. 48, no. 1/2, pp. 3–29, 2000.

[114] R. Huber and B. Kollmeier, "Pemo-qa new method for objective audio quality assessment using a model of auditory perception," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 6, pp. 1902–1911, 2006.

[115] I. Rec, "P. 863,perceptual objective listening quality assessment (polqa)," *Int. Telecomm. Union, Geneva*, 2011.

[116] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, vol. 2. IEEE, 2001, pp. 749–752.

[117] Y. Hu and P. C. Loizou, "Evaluation of objective quality measures for speech enhancement," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 16, no. 1, pp. 229–238, 2008.

[118] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time–frequency weighted noisy speech," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 7, pp. 2125–2136, 2011.

[119] J. Ma, Y. Hu, and P. C. Loizou, "Objective measures for predicting speech intelligibility in noisy conditions based on new band-importance functions," *The Journal of the Acoustical Society of America*, vol. 125, p. 3387, 2009.

[120] M. R. Schroeder, B. S. Atal, and J. Hall, "Optimizing digital speech coders by exploiting masking properties of the human ear," *The Journal of the Acoustical Society of America*, vol. 66, p. 1647, 1979.

[121] K. Hermus, W. Verhelst, P. Lemmerling, P. Wambacq, and S. Van Huffel, "Perceptual audio modeling with exponentially damped sinusoids," *Signal processing*, vol. 85, no. 1, pp. 163–176, 2005.

[122] D. De Koning and W. Verhelst, "On psychoacoustic noise shaping for audio requantization," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, vol. 5. IEEE, 2003, pp. V–453.

[123] M. D. Swanson, B. Zhu, A. H. Tewfik, and L. Boney, “Robust audio watermarking using perceptual masking,” *Signal Processing*, vol. 66, no. 3, pp. 337–355, 1998.

[124] N. Virag, “Single channel speech enhancement based on masking properties of the human auditory system,” *Speech and Audio Processing, IEEE Transactions on*, vol. 7, no. 2, pp. 126–137, 1999.

Part II

Precompensation Algorithms

Chapter 2

Hard Clipping Precompensation

Real-Time Perception-Based Clipping of Audio Signals Using Convex Optimization

Bruno Defraene, Toon van Waterschoot, Hans Joachim Ferreau,
Moritz Diehl, and Marc Moonen

Published[1] in *IEEE Trans. Audio Speech Language Process.*, vol. 20, no. 10, Dec. 2012, pp. 2657–2671.

[1] The text version included in this thesis manuscript includes minor adaptations in form and content with respect to the original published text.

Contributions of first author

- literature study
- co-design of perception-based clipping problem formulation
- co-development of dual active set optimization method
- co-development of projected gradient optimization methods
- co-design of evaluation experiments
- software implementation and computer simulations
- co-interpretation of simulation results
- co-formulation of conclusions
- text redaction and editing

Abstract

Clipping is an essential signal processing operation in many real-time audio applications, yet the use of existing clipping techniques generally has a detrimental effect on the perceived audio signal quality. In this paper, we present a novel multidisciplinary approach to clipping which aims to explicitly minimize the perceptible clipping-induced distortion by embedding a convex optimization criterion and a psychoacoustic model into a frame-based algorithm. The core of this *perception-based clipping* algorithm consists in solving a convex optimization problem for each time frame in a fast and reliable way. To this end, three different structure-exploiting optimization methods are derived in the common mathematical framework of convex optimization, and corresponding theoretical complexity bounds are provided. From comparative audio quality evaluation experiments, it is concluded that the perception-based clipping algorithm results in significantly higher objective audio quality scores than existing clipping techniques. Moreover, the algorithm is shown to be capable to adhere to real-time deadlines without making a sacrifice in terms of audio quality.

2.1 Introduction

In many real-time audio applications, the amplitude of a digital audio signal is not allowed to exceed a certain maximum level. This amplitude level restriction can be imposed for different generic or application-specific reasons. First, it can relate to an inherent limitation of the adopted digital representation of the signal. In this case, audio signal samples exceeding the allowable maximum amplitude level will either *wrap-around* or *saturate*, depending on the digital signal processing (DSP) system architecture [1]. In both modes, the result will be a significant degradation of the audio signal's sound quality. Secondly, the maximum amplitude level can be imposed in order to prevent the audio signal from exceeding the reproduction capabilities of the subsequent power amplifier and/or electroacoustic transducer stages. In fact, an audio signal exceeding this maximum amplitude level will not only result in a degradation of the sound quality of the reproduced audio signal (e.g. due to amplifier overdrive and loudspeaker saturation), but could possibly also damage the audio equipment. Thirdly, in music production applications, the amplitude level restriction is often set deliberately as part of a mastering/mixing process. Lastly, in hearing aid applications, the maximum amplitude level restriction is necessary to preserve a high listening comfort, as impulsive noises in the vicinity of the hearing aid user will sound uncomfortably loud if the audio signal amplitude is not properly limited.

In order to preserve a high sound quality of the reproduced audio signal and a high user listening comfort in the above mentioned applications, it is of paramount importance to instantaneously limit the digital audio signal with respect to the allowable maximum amplitude level. *Clippers* (or infinite limiters) are especially suited for this purpose: these alter incoming signal sample amplitudes such that no sample amplitude exceeds the maximum amplitude level (referred to as *clipping level* from here on) [2, Sec. 5.2]. Most existing clipping[2] techniques are governed by a static input-output characteristic, acting onto the input audio signal on a sample by sample basis by mapping a range of input amplitudes to a reduced range of output amplitudes. Depending on the sharpness of this input-output characteristic, one can distinguish between two types of clipping techniques: *hard clipping* and *soft clipping* [3], where the input-output characteristic exhibits an abrupt ("hard") or gradual ("soft") transition from the linear zone to the nonlinear zone respectively.

However, such a clipping operation itself introduces different kinds of unwanted distortion into the audio signal: odd harmonic distortion components, intermodulation distortion components and aliasing distortion components [4]. In a series of listening experiments performed on normal hearing subjects [5] and hearing-impaired subjects [6], it is concluded that the application of hard clipping and soft clipping to audio signals has a significant negative effect on perceptual sound quality scores, irrespective of the subject's hearing acuity. To our best knowledge, there have been no previous research efforts on improving the perceptual sound quality of existing clipping techniques. It is worthwhile to point out, however, recent research on the related problems of *audio declipping* and *audio imputation*, where the aim is to restore the missing values in clipped audio signals [7] [8].

In this paper, we propose a novel, multidisciplinary approach to clipping, aimed at minimizing the perceptible clipping-induced distortion. The proposed perception-based clipping algorithm combines aspects of digital signal processing, optimization theory and psychoacoustics. Hence, two algorithmic ingredients are novel compared to existing approaches:

- *Psychoacoustics*: incorporating knowledge about the human perception of sounds is indispensable for achieving minimally perceptible clipping-induced distortion. In other audio processing applications, the application of psychoacoustic principles and models has proven to be successful, e.g. in perceptual audio coding [9] and audio signal requantization [10].
- *Embedded convex optimization*: in an increasing number of signal processing applications, convex optimization is embedded directly into a signal processing algorithm in order to carry out nonlinear processing on the signal itself (as opposed to its more conventional use for e.g. linear filter design) [11]. In this framework, clipping of an audio signal will

[2] In this work, we use the word "clipping" to denote the deliberate operation of bounding the samples of a digital audio signal to a predefined maximum amplitude level. This should not be confused with the undesired "analog clipping phenomenon" as it can subsequently occur in various analog audio devices.

> be formulated as a sequence of constrained convex optimization problems regularly spaced in time, aimed at minimizing perceptible clipping-induced distortion. Real-time operation of such a scheme obviously calls for application-tailored optimization methods able to solve instances of the optimization problem at hand in a fast and reliable way. Therefore, we will spend extensive attention to three different structure-exploiting optimization methods and their comparative performance.

In previous work, a perception-based approach to clipping has been presented and was seen to significantly outperform existing clipping techniques in terms of objective sound quality scores [12]. This approach has been refined by incorporating a projected gradient optimization method for solving the constrained optimization problems under consideration [13]. In this paper, the main ideas presented in [12]-[13] will be reviewed and expanded, thereby introducing the following novel contributions:

- A new and significantly faster projected gradient optimization method will be proposed for solving the constrained optimization problems at hand, achieving an optimal linear convergence rate. By using this method, the perception-based clipping algorithm can effectively be applied in real time for very high solution accuracies.
- The different optimization methods will be rigorously described in the common mathematical framework of convex optimization. Advantages and disadvantages of the different optimization methods will be discussed, and theoretical complexity bounds will be derived in order to objectively compare their performance.
- The psychoacoustic principles and psychoacoustic model underpinning the perception-based clipping approach will be elaborated in detail.
- A thorough comparative objective perceptual evaluation of the proposed perception-based algorithm and existing clipping algorithms will be performed using two different objective measures of audio quality.

The paper is organized as follows. In Section 2.2, clipping is formulated as a sequence of constrained convex optimization problems, and the inclusion of a psychoacoustic model is discussed in detail. In Section 2.3, three different application-tailored convex optimization methods are proposed for solving the optimization problems at hand in a fast and reliable way, and corresponding theoretical complexity bounds are given. In Section 2.4, results are presented from a comparative audio quality evaluation of different clipping techniques, and an algorithmic complexity assessment of different optimization methods is performed. Finally, in Section 2.5, some concluding remarks are presented.

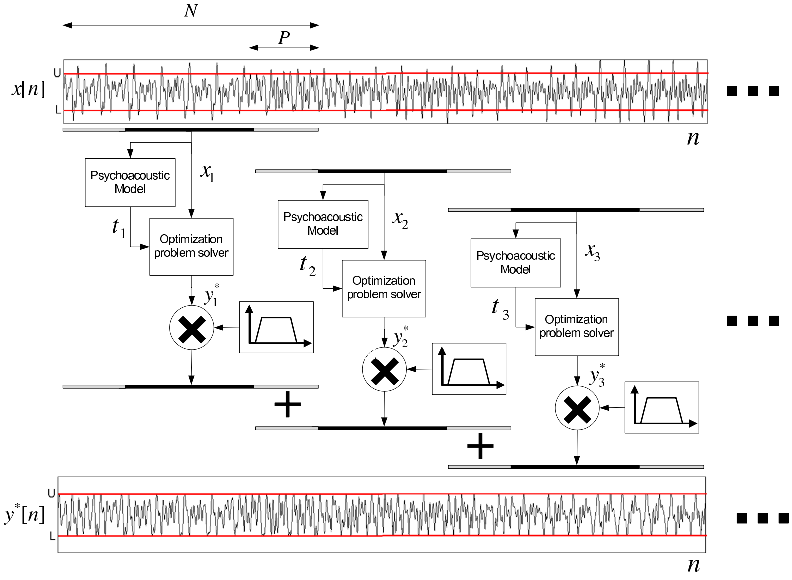


Figure 2.1: Schematic overview of the presented perception-based clipping technique

2.2 Perception-Based Clipping

2.2.1 General Description of the Algorithm

The goal of a clipping algorithm is to restrict the amplitude of a digital audio signal $x[n]$ to a given amplitude range $[L, U]$ (where $L \leq 0$ and $U \geq 0$), while keeping the clipped output signal $y[n]$ perceptually as close as possible to the input signal $x[n]$. In a perception-based clipping algorithm [12], the aim for *maximal perceptual similarity* is explicitly fulfilled, by

- incorporating into the algorithm knowledge about the human perception of sounds through the use of a psychoacoustic model.
- embedding into the algorithm the solution of an optimization problem, aimed at minimizing clipping-induced distortion.

Figure 2.1 schematically depicts the actual operation of the perception-based clipping algorithm presented in [12]. The digital input audio signal $x[n]$ is segmented into frames of N samples[3], with an overlap length of P samples between successive frames. The processing of one frame $\mathbf{x}_m$ consists of the following steps:

[3] Note that N is assumed an even number from here on.

1. Calculate the instantaneous global masking threshold $\mathbf{t}_m \in \mathbb{R}^{\frac{N}{2}+1}$ of the input frame $\mathbf{x}_m \in \mathbb{R}^N$, using part of the ISO/IEC 11172-3 MPEG-1 Layer 1 psychoacoustic model 1 [14]. The instantaneous global masking threshold of a signal gives the amount of distortion energy (dB) in each frequency bin that can be masked by the signal.
2. Calculate the optimal output frame $\mathbf{y}_m^* \in \mathbb{R}^N$ as the solution of a constrained optimization problem to be defined in subsection 2.2.2.
3. Apply a trapezoidal window to the optimal output frame $\mathbf{y}_m^*$ and sum the optimal output frames to form a continuous output audio signal $y^*[n]$.

2.2.2 Optimization Problem Formulation

The core of the perception-based clipping technique consists in calculating the solution of a constrained optimization problem for each frame. From the knowledge of the input frame $\mathbf{x}_m$ and its instantaneous properties, the optimal output frame $\mathbf{y}_m^*$ is calculated. Let us define the optimization variable of the problem as $\mathbf{y}_m$, i.e. the output frame. A necessary constraint on the output frame $\mathbf{y}_m$ is that the amplitude of the output samples cannot exceed the upper and lower clipping levels U and L. The objective function f we want to minimize must reflect the amount of perceptible distortion added between $\mathbf{y}_m$ and $\mathbf{x}_m$. We can thus fomulate the optimization problem as an inequality constrained frequency domain weighted L2-distance minimization, i.e.

$$\begin{aligned}\mathbf{y}_m^* &= \underset{\mathbf{y}_m \in \mathbb{R}^N}{\arg\min}\ f(\mathbf{y}_m) \quad \text{s.t.} \quad \begin{cases} h_i(\mathbf{y}_m) = U - y_{m,i} \geq 0 \\ h_{N+i}(\mathbf{y}_m) = y_{m,i} - L \geq 0 \end{cases} \quad , i = 0, ..., N-1 \\ &= \underset{\mathbf{y}_m \in \mathbb{R}^N}{\arg\min}\ \frac{1}{2N} \sum_{i=0}^{N-1} w_{m,i} |Y_m(e^{j\omega_i}) - X_m(e^{j\omega_i})|^2 \quad \text{s.t.} \quad \mathbf{l} \leq \mathbf{y}_m \leq \mathbf{u} \quad (2.1)\end{aligned}$$

where $\omega_i = (2\pi i)/N$ represents the discrete frequency variable, $X_m(e^{j\omega_i})$ and $Y_m(e^{j\omega_i})$ are the discrete frequency components of $\mathbf{x}_m$ and $\mathbf{y}_m$ respectively, the vectors $\mathbf{u} = U\mathbb{1}_N$ and $\mathbf{l} = L\mathbb{1}_N$ contain the upper and lower clipping levels respectively (with $\mathbb{1}_N \in \mathbb{R}^N$ an all ones vector), and $w_{m,i}$ are the weights of a perceptual weighting function to be defined in subsection 2.2.3. Notice that in case the input frame $\mathbf{x}_m$ does not violate the inequality constraints, the optimization problem (2.1) trivially has the solution $\mathbf{y}_m^* = \mathbf{x}_m$ and the input frame is transmitted unaltered by the clipping algorithm.

Formulation (2.1) of the optimization problem can be rewritten as follows [4]

$$\mathbf{y}_m^* = \underset{\mathbf{y}_m \in \mathbb{R}^N}{\arg\min}\ \frac{1}{2} (\mathbf{y}_m - \mathbf{x}_m)^H\ \mathbf{D}^H \mathbf{W}_m \mathbf{D}\ (\mathbf{y}_m - \mathbf{x}_m) \quad \text{s.t.} \quad \mathbf{l} \leq \mathbf{y}_m \leq \mathbf{u}$$

[4] In this text, the superscripts T and H denote the transpose and the Hermitian transpose, respectively.

$$= \arg\min_{\mathbf{y}_m \in \mathbb{R}^N} \frac{1}{2}\, \mathbf{y}_m^H \underbrace{\mathbf{D}^H \mathbf{W}_m \mathbf{D}}_{\triangleq \mathbf{H}_m} \mathbf{y}_m + (\underbrace{-\mathbf{D}^H \mathbf{W}_m \mathbf{D}\, \mathbf{x}_m}_{\triangleq \mathbf{g}_m = -\mathbf{H}_m \mathbf{x}_m})^H \mathbf{y}_m \quad \text{s.t.} \quad \mathbf{l} \leq \mathbf{y}_m \leq \mathbf{u} \tag{2.2}$$

where $\mathbf{D} \in \mathbb{C}^{N \times N}$ is the unitary Discrete Fourier Transform (DFT) matrix defined as

$$\mathbf{D} = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & e^{-j\omega_1} & e^{-j\omega_2} & \dots & e^{-j\omega_{N-1}} \\ 1 & e^{-j\omega_2} & e^{-j\omega_4} & \dots & e^{-j\omega_{2(N-1)}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & e^{-j\omega_{N-1}} & e^{-j\omega_{2(N-1)}} & \dots & e^{-j\omega_{(N-1)(N-1)}} \end{bmatrix} \tag{2.3}$$

and $\mathbf{W}_m \in \mathbb{R}^{N \times N}$ is a diagonal weighting matrix with positive weights $w_{m,i}$, obeing the symmetry property $w_{m,i} = w_{m,N-i}$ for $i = 1, 2, ..., \frac{N}{2} - 1$,

$$\mathbf{W}_m = \begin{bmatrix} w_{m,0} & 0 & 0 & \dots & 0 \\ 0 & w_{m,1} & 0 & \dots & 0 \\ 0 & 0 & w_{m,2} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & w_{m,N-1} \end{bmatrix} \tag{2.4}$$

We remark that the objective function in (2.2) is a quadratic function and that the constraint functions are affine, hence optimization problem (2.2) constitutes a *quadratic program* (QP). Note also that the choice for a quadratic error in the objective function was made in order to strike a balance between perceptual relevance on one hand, and mathematical elegance and suitability of the objective function from an optimization point of view on the other hand. With this trade-off in mind, the use of a quadratic error criterion was preferred over other considered alternatives.

The number of samples N per audio signal frame is defined by the sampling rate of the audio signal (typically ranging from 16 kHz to 44.1 kHz) and the time duration of the audio signal frame (typically ranging from 10 to 30 ms). For most audio applications, the dimension N of the per-frame QP is thus expected to lie within the range of 100 to 1000 optimization variables.

2.2.3 Perceptual Weighting Function

In order for the objective function in optimization problem (2.1) to reflect the amount of perceptible distortion added between input frame $\mathbf{x}_m$ and output frame $\mathbf{y}_m$, the perceptual weighting function $\mathbf{w}_m$ must be constructed judiciously. The rationale behind applying signal-dependent weights in the summation (2.1) is the psychoacoustic fact that distortion at certain frequencies

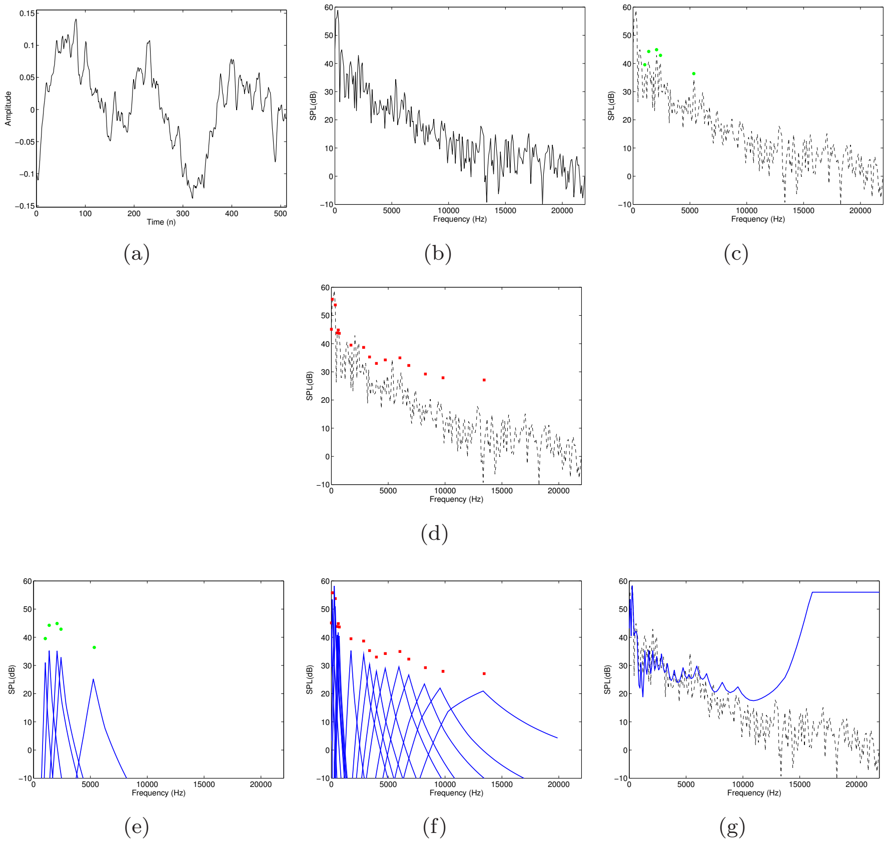


Figure 2.2: Different steps in the computation of the global masking threshold using the ISO/IEC 11172-3 MPEG-1 Layer 1 psychoacoustic model 1 : (a)-(b) Time domain and normalized frequency domain representations of the input audio signal (c)-(d) Tonal maskers (circles), non-tonal maskers (squares) and input frequency spectrum (dotted line) (e)-(f) Individual masking thresholds related to tonal and non-tonal maskers respectively (g) Global masking threshold (solid line) and input frequency spectrum (dotted line)

is more perceptible than distortion at other frequencies, and that the relative perceptibility is mostly signal-dependent. Two phenomena of human auditory perception are responsible for this,

- The *absolute threshold of hearing* is defined as the required intensity (dB) of a pure tone such that an average listener will just hear the tone in a noiseless environment. The absolute threshold of hearing is a function of the tone frequency and has been measured experimentally [15].
- *Simultaneous masking* is a phenomenon where the presence of certain spectral energy (the masker) masks the simultaneous presence of weaker spectral energy (the maskee), or in other words, renders it imperceptible.

Combining both phenomena, the instantaneous global masking threshold of a signal gives the amount of distortion energy (dB) at each frequency bin that can be masked by the signal. In this framework, we consider the input frame $\mathbf{x}_m$ to act as the masker, and $\mathbf{y}_m - \mathbf{x}_m$ as the maskee. In other words, we make the assumption that when the ear is presented with the output frame $\mathbf{y}_m$, it is in fact presented with the input frame $\mathbf{x}_m$ and the distortion $\mathbf{y}_m - \mathbf{x}_m$ simultaneously, and that the simultaneous masker-maskee relationship between both signals can be exploited. By selecting the weights $w_{m,i}$ in (2.1) to be exponentially decreasing with the value of the global masking threshold of $\mathbf{x}_m$ at frequency bin i, the objective function effectively reflects the amount of perceptible distortion introduced[5]. This can be specified as

$$w_{m,i} = \begin{cases} 10^{-\alpha t_{m,i}} & \text{if } 0 \leq i \leq \frac{N}{2} \\ 10^{-\alpha t_{m,N-i}} & \text{if } \frac{N}{2} < i \leq N-1 \end{cases} \tag{2.5}$$

where $\mathbf{t}_m$ is the global masking threshold of $\mathbf{x}_m$ (in dB). Appropriate values of the compression parameter α have been determined to lie in the range 0.04-0.06.

The instantaneous global masking threshold $\mathbf{t}_m$ of an input frame $\mathbf{x}_m$ is calculated using part of the ISO/IEC 11172-3 MPEG-1 Layer 1 psychoacoustic model 1. A complete description of the operation of this psychoacoustic model is out of the scope of this paper (we refer the reader to [9] and [14]). We will outline the relevant steps in the computation of the instantaneous global masking threshold and illustrate the result of each step on an example audio frame (see Figure 2.2):

1. *Spectral analysis and SPL normalization*: In this step a high-resolution spectral estimate of the input frame is calculated, with spectral components expressed in terms of sound pressure level (SPL). After a normalization operation and a Hann window operation on the input signal frame, the PSD estimate is obtained through a 512-point Fast Fourier Transform (FFT). Figure 2.2(a) shows the time-domain input signal, Figure 2.2(b) shows the resulting spectral estimate.

[5] Indeed, the terms in the summation of objective function (2.1) can then be seen to resemble to distortion-to-masker power ratios.

2. *Identification of tonal and non-tonal maskers*: It is known from psychoacoustic research that the tonality of a masking component has an influence on its masking properties [16]. For this reason it is important to discriminate between tonal maskers (defined as local maxima of the signal spectrum) and non-tonal maskers. The output of the FFT is used to determine the relevant tonal and non-tonal maskers in the spectrum of the audio signal. In a first phase, tonal maskers are identified at local maxima of the PSD: energy from three adjacent spectral components centered at the local maximum is combined to form a single tonal masker. In a second phase, a single non-tonal masker per critical band is formed by addition of all the energy from the spectral components within the critical band that have not contributed to a tonal masker.
3. *Decimation of maskers*: In this step, the number of maskers is reduced using two criteria. First, any tonal or non-tonal masker below the absolute threshold of hearing is discarded. Next, any pair of maskers occurring within a distance of 0.5 Bark is replaced by the stronger of the two. Figures 2.2(c) and 2.2(d) respectively depict the identified tonal and non-tonal maskers, after decimation.
4. *Calculation of individual masking thresholds*: an individual masking threshold is calculated for each masker in the decimated set of tonal and non-tonal maskers, using fixed psychoacoustic rules. Essentially, the individual masking threshold depends on the frequency, loudness level and tonality of the masker. Figure 2.2(e) and 2.2(f) show the individual masking thresholds associated with tonal and non-tonal maskers, respectively.
5. *Calculation of global masking threshold*: Finally, the global masking threshold is calculated by a power-additive combination of the tonal and non-tonal individual masking thresholds, and the absolute threshold of hearing. This is illustrated in Figure 2.2(g).

2.3 Optimization Methods

The core of the perception-based clipping algorithm described in Section 2.2 is formed by the solution of an instance of optimization problem (2.2) for every frame $\mathbf{x}_m$. Looking at the relatively high sampling rates (e.g. 44.1 kHz for CD-quality audio) and associated frame rates under consideration, it is clear that real-time operation of the algorithm calls for application-tailored optimization methods to solve the optimization problems in a fast and reliable way. In this section, we will discuss three different structure-exploiting optimization methods, whose common ground is the notion of *convex optimization*. Before doing so, we will first look at the properties of the optimization problem at hand in this framework of convex optimization.

2.3.1 Convex Optimization Framework

Convex optimization is a subfield of mathematical optimization that studies a special class of mathematical optimization problems, namely convex optimization problems. This class can be formally defined as follows.

Definition 2.1 [Convex optimization problem] A convex optimization problem is one of the form

$$\begin{aligned} \min_{\mathbf{y}\in\mathbb{R}^N} f(\mathbf{y}) \quad \text{s.t.} \quad & h_i(\mathbf{y}) \leq 0, \quad i = 0, ..., K-1 \\ & \mathbf{c}_j^T\mathbf{y} = \mathbf{d}_j, \quad j = 0, ..., M-1 \end{aligned} \tag{2.6}$$

in which the objective function f and the constraint functions $h_0, ..., h_{K-1}$ are convex, which means they satisfy

$$f(a\mathbf{x} + b\mathbf{y}) \leq af(\mathbf{x}) + bf(\mathbf{y}) \tag{2.7}$$

$$h_i(a\mathbf{x} + b\mathbf{y}) \leq ah_i(\mathbf{x}) + bh_i(\mathbf{y}) \tag{2.8}$$

for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^N$ and all $a, b \in \mathbb{R}$ with $a + b = 1$, $a \geq 0$, $b \geq 0$. □

In particular, for quadratic programs, the next definition holds.

Definition 2.2 [Convex QP] A quadratic program (QP) is *convex* if and only if the Hessian matrix $\nabla^2 f(\mathbf{y})$ is positive semi-definite. It is *strongly convex* if and only if $\nabla^2 f(\mathbf{y})$ is positive definite. □

A fundamental property of convex optimization problems is that any local minimizer is a global minimizer. For strongly convex problems, it will also be the unique minimizer.

We will now show that optimization problem (2.2) is a convex quadratic program, thereby looking a bit deeper into the structure of the Hessian matrix $\mathbf{H}_m$.

Definition 2.3 [Circulant matrix] A circulant matrix $\mathbf{C}$ is a square matrix having the form

$$\mathbf{C} = \begin{bmatrix} c_0 & c_{N-1} & & \cdots & & c_1 \\ c_1 & c_0 & c_{N-1} & & & \vdots \\ c_2 & c_1 & c_0 & \ddots & & \\ & c_2 & c_1 & \ddots & & c_{N-1} \\ \vdots & & & \ddots & & \\ c_{N-1} & \cdots & & c_2 & c_1 & c_0 \end{bmatrix} \tag{2.9}$$

where each row is a cyclic shift of the row above it. □

Theorem 2.4 (Diagonalization of circulant matrices [17, 18]) *A circulant matrix* $\mathbf{C} \in \mathbb{C}^{N \times N}$ *is diagonalized by the unitary DFT matrix* $\mathbf{D}$ *defined in* (2.3)*, i.e.*

$$\mathbf{C} = \mathbf{D}^H \mathbf{\Psi} \mathbf{D} \tag{2.10}$$

where $\mathbf{\Psi} = \mathrm{diag}(\boldsymbol{\psi})$ *contains the eigenvalues of* $\mathbf{C}$*, which are obtained as the DFT of the first column of* $\mathbf{C}$*,*

$$\psi_l = \sum_{k=0}^{N-1} c_k e^{-j\omega_{lk}}, \quad l = 0, ..., N-1 \tag{2.11}$$

We can now state the following important properties of the Hessian matrix $\mathbf{H}_m = \mathbf{D}^H \mathbf{W}_m \mathbf{D}$ in (2.2):

Theorem 2.5 *The Hessian matrix* $\mathbf{H}_m = \mathbf{D}^H \mathbf{W}_m \mathbf{D} \in \mathbb{R}^{N \times N}$ *in optimization problem* (2.2) *is real, symmetric, positive definite and circulant.*

Proof : From (2.10) in Theorem 2.4, we readily see that $\mathbf{H}_m$ is *circulant* and $\mathbf{W}_m$ contains its eigenvalues. By definitions (2.4)-(2.5), the elemens of $\mathbf{W}_m$ are real and have even symmetry, so by (2.11) the first column of $\mathbf{H}_m$ will also be real and have even symmetry. From this we can see that $\mathbf{H}_m$ is *real* and *symmetric*. As a symmetric matrix is positive definite if and only if all of its eigenvalues are positive, it remains to remark that the elements of $\mathbf{W}_m$ are positive by construction (2.5) to conclude that $\mathbf{H}_m$ is *positive definite*. □

Corollary 2.6 *Optimization problem* (2.2) *is a strongly convex quadratic program.*

Convex optimization problems can be solved reliably and efficiently by using special methods for convex optimization. In particular, different iterative optimization methods for solving (strongly) convex QPs have been presented. Essentially, three classes of methods can be distinguished. A first class are the *projected gradient methods*, where only first-order information is used for solving the optimization problem. These methods are conceptually simple and computationally cheap, but typically suffer from slow convergence, sometimes preventing their use in real-time applications. However, it is generally possible to perform a convergence analysis and to establish useful polynomial computational complexity bounds for these algorithms [19]. A second class are *interior-point methods*: these rely on heavier computational tasks, but have a better convergence rate. Some interior-point methods are polynomial time, but the complexity bounds are generally far off from practically observed ones [20]. A third class are *active set methods*: these have a good perfomance in practice, but suffer from the drawback that in general no polynomial complexity bounds can be given [21].

In the remainder of this section, we will propose three different optimization methods tailored to QP (2.2), wherein the structure of the optimization problem will be exploited.

- In subsection 2.3.2, an active set type of method will be proposed which exploits the fact that only a small subset of the constraints will influence the final solution.
- In subsections 2.3.3 and 2.3.4, two projected gradient methods will be proposed which exploit the circulant structure of the Hessian matrix and the geometry of the convex feasible set.

2.3.2 Optimization Method 1: Dual Active Set Strategy

In [12], it was experimentally shown that general-purpose QP solvers are largely inadequate to solve instances of QP (2.2) in real time. Therefore, an active set optimization method was proposed that efficiently solves the *dual optimization problem* of (2.2).

Definition 2.7 [Dual optimization problem] For any primal optimization problem of the form (2.6), the dual optimization problem is defined as the convex maximization problem

$$\underbrace{\max_{\boldsymbol{\lambda}\in\mathbb{R}^K,\boldsymbol{\nu}\in\mathbb{R}^M} \inf_{\mathbf{y}\in\mathbb{R}^N} \left[\underbrace{f(\mathbf{y}) - \sum_{i=0}^{K-1} \lambda_i h_i(\mathbf{y}) - \sum_{j=0}^{M-1} \nu_j(\mathbf{c}_j^T\mathbf{y} - \mathbf{d}_j)}_{\triangleq \mathcal{L}(\mathbf{y},\boldsymbol{\lambda},\boldsymbol{\nu})} \right]}_{\triangleq q(\boldsymbol{\lambda},\boldsymbol{\nu})} \quad \text{s.t.} \quad \boldsymbol{\lambda} \geq \mathbf{0} \tag{2.12}$$

where $\boldsymbol{\lambda}$ and $\boldsymbol{\nu}$ are the vectors of *Lagrange multipliers* associated to the inequality constraints and equality constraints respectively, $\mathcal{L}(\mathbf{y},\boldsymbol{\lambda},\boldsymbol{\nu})$ is the *Lagrangian function* and $q(\boldsymbol{\lambda},\boldsymbol{\nu})$ is the *Lagrange dual function.* □

A primal optimization problem and its associated dual optimization problem are related in interesting ways. In general, the dual optimization problem can be used to obtain a lower bound on the optimal value of the objective for the primal problem. This primal-dual relationship is known as *weak duality* [22].

Theorem 2.8 (Weak duality) *The optimal objective value p^* of any primal optimization problem* (2.6) *and the optimal objective value d^* of the associated* *dual optimization problem* (2.12) *are related as follows,*

$$d^* \leq p^* \tag{2.13}$$

One of the main advantages of convex optimization problems is the fact that the dual optimization problem can be used directly for solving the primal optimization problem. This primal-dual relationship is known as *strong duality* [22].

Theorem 2.9 (Strong duality) *If the primal optimization problem* (2.6) *is convex and it has a strictly feasible point, then primal optimization problem* (2.6) *and dual optimization problem* (2.12) *have the same optimal objective value,*

$$d^* = p^* \tag{2.14}$$

Since in subsection 2.3.1 optimization problem (2.2) was shown to be a convex quadratic program, and zero is a strictly feasible point, it is clear from Theorem 2.9 that it has a strong duality relationship with its dual counterpart. We formulate this dual optimization problem as follows. First, the Lagrangian function $\mathcal{L}(\mathbf{y}_m, \boldsymbol{\lambda}_{m,u}, \boldsymbol{\lambda}_{m,l})$ is given by

$$\begin{aligned}\mathcal{L}(\mathbf{y}_m, \boldsymbol{\lambda}_{m,u}, \boldsymbol{\lambda}_{m,l}) &= f(\mathbf{y}_m) - \boldsymbol{\lambda}_{m,u}^T \mathbf{h}_u(\mathbf{y}_m) - \boldsymbol{\lambda}_{m,l}^T \mathbf{h}_l(\mathbf{y}_m) \\ &= \frac{1}{2}\left(\mathbf{y}_m - \mathbf{x}_m\right)^T \mathbf{H}_m \left(\mathbf{y}_m - \mathbf{x}_m\right) + \boldsymbol{\lambda}_{m,u}^T(\mathbf{y}_m - \mathbf{u}) + \boldsymbol{\lambda}_{m,l}^T(\mathbf{l} - \mathbf{y}_m)\end{aligned} \tag{2.15}$$

where $\boldsymbol{\lambda}_{m,u}, \boldsymbol{\lambda}_{m,l} \in \mathbb{R}^N$ denote the vectors of Lagrange multipliers associated to the upper clipping level constraints $\mathbf{h}_u(\mathbf{y}_m) = [h_0(\mathbf{y}_m) \dots h_{N-1}(\mathbf{y}_m)]^T$ and the lower clipping level constraints $\mathbf{h}_l(\mathbf{y}_m) = [h_N(\mathbf{y}_m) \dots h_{2N-1}(\mathbf{y}_m)]^T$, respectively. Then, the Lagrange dual function equals

$$\begin{aligned}q(\boldsymbol{\lambda}_{m,u}, \boldsymbol{\lambda}_{m,l}) &= \inf_{\mathbf{y}_m \in \mathbb{R}^N} \mathcal{L}(\mathbf{y}_m, \boldsymbol{\lambda}_{m,u}, \boldsymbol{\lambda}_{m,l}) \\ &= -\frac{1}{2}(\boldsymbol{\lambda}_{m,u} - \boldsymbol{\lambda}_{m,l} - \mathbf{H}_m\mathbf{x}_m)^T \mathbf{H}_m^{-1} (\boldsymbol{\lambda}_{m,u} - \boldsymbol{\lambda}_{m,l} - \mathbf{H}_m\mathbf{x}_m) \\ &\qquad - \boldsymbol{\lambda}_{m,u}^T\mathbf{u} + \boldsymbol{\lambda}_{m,l}^T\mathbf{l} + \frac{1}{2}\,\mathbf{x}_m^T\mathbf{H}_m\mathbf{x}_m\end{aligned} \tag{2.16}$$

Finally, the dual optimization problem can be formulated as

$$\begin{aligned}\boldsymbol{\lambda}_m^* &= \arg\max_{\boldsymbol{\lambda}_m \in \mathbb{R}^{2N}} \; q(\boldsymbol{\lambda}_m) \quad \text{s.t. } \boldsymbol{\lambda}_m \geq \mathbf{0} \\ &= \arg\min_{\boldsymbol{\lambda}_m \in \mathbb{R}^{2N}} \; \frac{1}{2}\boldsymbol{\lambda}_m^T \underbrace{\mathbf{A}^T\mathbf{H}_m^{-1}\mathbf{A}}_{\triangleq \tilde{\mathbf{H}}_m} \boldsymbol{\lambda}_m + (\underbrace{\mathbf{B}^T\mathbb{1}_N - \mathbf{A}^T\mathbf{x}_m}_{\triangleq \tilde{\mathbf{g}}_m})^T \boldsymbol{\lambda}_m \quad \text{s.t.} \quad \boldsymbol{\lambda}_m \geq \mathbf{0}\end{aligned} \tag{2.17}$$

where $\boldsymbol{\lambda}_m \in \mathbb{R}^{2N}$, $\mathbf{A} \in \mathbb{R}^{N \times 2N}$ and $\mathbf{B} \in \mathbb{R}^{N \times 2N}$ are defined as $\boldsymbol{\lambda}_m = \left[\begin{array}{cc} \boldsymbol{\lambda}_{m,u}^T & \boldsymbol{\lambda}_{m,l}^T \end{array} \right]^T$, $\mathbf{A} = \left[\begin{array}{c|c} \mathbf{I}_N & -\mathbf{I}_N \end{array} \right]$, and $\mathbf{B} = \left[\begin{array}{c|c} U\mathbf{I}_N & -L\mathbf{I}_N \end{array} \right]$, with $\mathbf{I}_N \in \mathbb{R}^{N \times N}$ the identity matrix.

Computation of $\mathbf{y}_m^*$ from $\boldsymbol{\lambda}_m^*$ is then straightforward,

$$\begin{aligned} \mathbf{y}_m^* &= -\mathbf{H}_m^{-1}(\mathbf{A}\boldsymbol{\lambda}_m^* - \mathbf{H}_m\mathbf{x}_m) \\ &= \mathbf{x}_m - \mathbf{H}_m^{-1}\mathbf{A}\boldsymbol{\lambda}_m^* \end{aligned} \tag{2.18}$$

The dual optimization problem formulated in (2.17)-(2.18) can be solved efficiently by exploiting the fact that only a small subset of the large number ($2N$) of inequality constraints are expected to influence the solution. Under the assumption of a moderate clipping factor, indeed only a small number of samples in the input frame $\mathbf{x}_m$ are expected to exceed the clipping levels. An iterative external active set strategy is adopted, where the following steps are executed in each iteration (see Algorithm 1)[6] :

1. Check which inequality constraints are violated in the previous solution iterate. In case no inequality constraints are violated, the algorithm terminates.
2. Add the violated constraints to an active set $\mathbb{S}$ of constraints to be monitored.
3. Solve a small-scale QP corresponding to (2.17) with those Lagrange multipliers corresponding to constraints not in the active set $\mathbb{S}$ set to zero.
4. Compute the new solution iterate by evaluating (2.18).

Using this strategy of dualizing and iteratively adapting an appropriate subset of inequality constraints, a QP dimensionality reduction is achieved which brings along a significant computational complexity reduction. In effect, the solution of QP (2.2) is found by solving a small number of small scale QPs (2.17) instead of by solving the full scale QP at once. From simulations, it is concluded that 4 external iterations generally suffice for solving an instance of optimization problem (2.2). In comparison to general-purpose dense QP solvers, the method achieves a reduction of computation time with a factor ranging from 10 up to 200. Moreover, for clipping factors[7] higher than 0.95, the method could potentially be used in a real-time clipping context [12].

Although computation times are reduced considerably, it has to be remarked that this optimization method still has a few shortcomings, possibly preventing it to be used reliably in real-time audio applications:

- First of all, the method's computational complexity is seen to increase with

[6] We introduce in this algorithm the notation $[\cdot]_m^k$ for the kth iterate of the mth frame

[7] Clipping factor CF is defined as 1-(fraction of signal samples exceeding the upper or lower clipping level)

Algorithm 1 Dual active set strategy

```
Input $\mathbf{x}_m \in \mathbb{R}^N$, $\mathbf{W}_m$, $L$, $U$
Output $\mathbf{y}_m^* \in \mathbb{R}^N$
$\mathbf{y}_m^0 = \mathbf{x}_m$
$\mathbb{V}_m^0 = \{i | h_i(\mathbf{y}_m^0) < 0\}$
$\mathbb{S}_m^0 = \mathbb{V}_m^0$
$k = 0$
while $\mathbb{V}_m^k \neq \emptyset$ do
    $\lambda_{m,i}^{k+1} = 0,\ i \notin \mathbb{S}_m^k$
    Calculate $\lambda_{m,i}^{k+1},\ i \in \mathbb{S}_m^k$ as solution of small-scale QP defined in (2.17)
    $\mathbf{y}_m^{k+1} = \mathbf{x}_m - \mathbf{H}_m^{-1}\mathbf{A}\boldsymbol{\lambda}_m^{k+1}$ [using (2.18)]
    Collect index set of violated inequality constraints $\mathbb{V}_m^{k+1} = \{i | h_i(\mathbf{y}_m^{k+1}) < 0\}$
    Extend active set $\mathbb{S}_m^{k+1} = \mathbb{S}_m^k \cup \mathbb{V}_m^{k+1}$
    $k = k + 1$
end while
$\mathbf{y}_m^* = \mathbf{y}_m^k$
```

increasing number of violated constraints in the input frame $\mathbf{x}_m$. That is, the computational complexity increases with decreasing clipping factors, making it impossible to run the method in real time for low clipping factors.

- Secondly, it is practically impossible to derive certifying polynomial complexity bounds for the optimization method.
- Lastly, the iterative optimization method cannot be stopped early (i.e. before convergence to the exact solution) to provide an approximate solution of the optimization problem.

2.3.3 Optimization Method 2: Projected Gradient Descent

In this subsection, we present a projected gradient optimization method that deals with the different issues raised in subsection 2.3.2 concerning applicability in real time. First, a general description of the method is given. Then, the selection of an appropriate stepsize is discussed. Finally, the computation of approximate solutions is discussed and theoretical algorithmic complexity bounds are derived.

Description of the Method

Projected gradient methods are a class of iterative methods for solving optimization problems over convex sets. In each iteration, first a step is taken along the negative gradient direction of the objective function, after which the result is orthogonally projected onto the convex feasible set, thereby maintaining feasibility of the iterates [23]. A low computational complexity per iteration is the main asset of projected gradient methods, provided that the orthogonal projection onto the convex feasible set and the gradient of the objective function can easily be computed.

For optimization problem (2.2), both these elements can indeed be computed at an extremely low computational complexity, by exploiting the structure of the Hessian matrix and the convex feasible set. The main steps to be performed in the $(k+1)$th iteration of the proposed projected gradient method are as follows:

- Take a step with stepsize s_m^k along the negative gradient direction:

  $$\tilde{\mathbf{y}}_m^{k+1} = \mathbf{y}_m^k - s_m^k \nabla f(\mathbf{y}_m^k) \tag{2.19}$$

  where using (2.2),

  $$\begin{aligned}\nabla f(\mathbf{y}_m^k) &= \mathbf{H}_m(\mathbf{y}_m^k - \mathbf{x}_m) \\ &= \mathbf{D}^H \mathbf{W}_m \mathbf{D}(\mathbf{y}_m^k - \mathbf{x}_m).\end{aligned} \tag{2.20}$$

  It is clear from (2.20) that the gradient computation can be performed at a very low computational complexity, by sequentially applying a DFT (multiplication by $\mathbf{D}$), an element-wise weighting (multiplication by $\mathbf{W}_m$), and an IDFT (multiplication by $\mathbf{D}^H$) to the vector $(\mathbf{y}_m^k - \mathbf{x}_m)$. An alternative interpretation is that we perform a matrix-vector multiplication of the circulant matrix $\mathbf{H}_m$ with the vector $(\mathbf{y}_m^k - \mathbf{x}_m)$. By exploiting the computational efficiency of the FFT algorithm, the gradient computation thus has a complexity of $O(N \log_2 N)$.

- Project $\tilde{\mathbf{y}}_m^{k+1}$ orthogonally onto the convex feasible set Q of (2.2), which is defined as

  $$Q = \{\mathbf{y}_m \in \mathbb{R}^N \,|\, \mathbf{l} \leq \mathbf{y}_m \leq \mathbf{u}\} \tag{2.21}$$

  The feasible set can be thought of as an N-dimensional box. An orthogonal projection $\Pi_Q(\tilde{\mathbf{y}}_m^{k+1})$ onto this N-dimensional box boils down to performing a simple componentwise hard clipping operation (with lower bound L and upper bound U), i.e.

  $$\mathbf{y}_m^{k+1} = \Pi_Q(\tilde{\mathbf{y}}_m^{k+1}) = \arg\min_{\mathbf{z} \in Q} \frac{1}{2} \|\mathbf{z} - \tilde{\mathbf{y}}_m^{k+1}\|_2^2 \tag{2.22}$$

  where

  $$y_{m,i}^{k+1} = \begin{cases} L & \text{if} \quad \tilde{y}_{m,i}^{k+1} < L \\ \tilde{y}_{m,i}^{k+1} & \text{if} \quad L \leq \tilde{y}_{m,i}^{k+1} \leq U \\ U & \text{if} \quad \tilde{y}_{m,i}^{k+1} > U \end{cases} \quad , \; i = 0...N-1 \tag{2.23}$$

Stepsize Selection

Several rules for selecting stepsizes s_m^k in projected gradient methods have been proposed in literature, e.g. fixed stepsizes, diminishing stepsizes, or line search rules [23]. Here a fixed stepsize is used, thereby avoiding the additional computational complexity incurred by line searches. In [19], it is shown that by choosing a fixed stepsize

$$s_m^k = \frac{1}{C_m}, \ \forall k \geq 0 \tag{2.24}$$

with C_m the Lipschitz constant of the gradient ∇f of (2.1) on the set Q (for frame m), a limit point of the sequence $\{\mathbf{y}_m^k\}$ obtained by iteratively applying (2.19) and (2.23) is a stationary point. Because of the convexity of f, it is a local minimum and hence a global minimum.

In order to establish the Lipschitz constant C_m of our problem, we introduce the next lemma.

Lemma 1 *Let function f be twice continuously differentiable on set Q. The gradient ∇f is Lipschitz continuous on set Q with Lipschitz constant C if and only if*

$$||\nabla^2 f(\mathbf{z})|| \leq C \ , \ \forall \mathbf{z} \in Q \tag{2.25}$$

In other words, the Lipschitz constant of the gradient ∇f can be seen as an upper bound to the curvature of the objective function f. Using this lemma, we can easily show that the Lipschitz constant C_m is computed as

$$\begin{aligned} C_m &= ||\mathbf{H}_m|| \\ &= \max_{1 \leq i \leq N} \beta_i(\mathbf{H}_m) \\ &= \max_{1 \leq i \leq N} \beta_i(\mathbf{D}^H \mathbf{W}_m \mathbf{D}) \\ &= \max_{0 \leq i \leq N-1} w_{m,i} \end{aligned} \tag{2.26}$$

where $\beta_i(\mathbf{H}_m)$, $i = 1...N$, denote the eigenvalues of the Hessian matrix $\mathbf{H}_m$.

Strong convexity assumes that in addition to an upper bound also a lower bound on the curvature of the objective function can be found, determined by the *convexity parameter* μ_m.

Lemma 2 *Let function f be twice continuously differentiable on set Q. The function f is strongly convex on set Q with convexity parameter μ if and only if there exists $\mu > 0$ such that*

$$\nabla^2 f(\mathbf{z}) \geq \mu \mathbf{I} \ , \ \forall \mathbf{z} \in Q$$

Algorithm 2 Projected gradient method

```
Input $\mathbf{x}_m \in \mathbb{R}^N$, $\mathbf{y}_m^0 \in Q$, $\mathbf{W}_m$, $L$, $U$,
Output $\mathbf{y}_m^* \in \mathbb{R}^N$
$k = 0$
Calculate Lipschitz constant $C_m$ [using (2.26)]
while convergence is not reached do
    $\tilde{\mathbf{y}}_m^{k+1} = \mathbf{y}_m^k - \frac{1}{C_m}\nabla f(\mathbf{y}_m^k)$    [using (2.20)]
    $\mathbf{y}_m^{k+1} = \Pi_Q(\tilde{\mathbf{y}}_m^{k+1})$    [using (2.23)]
    $k = k + 1$
end while
$\mathbf{y}_m^* = \mathbf{y}_m^k$
```

Using the former lemma, we can prove that for our objective function the convexity parameter μ_m can be computed as

$$\begin{aligned}\mu_m &= \min_{1 \le i \le N} \beta_i(\mathbf{H}_m) \\ &= \min_{1 \le i \le N} \beta_i(\mathbf{D}^H \mathbf{W}_m \mathbf{D}) \\ &= \min_{0 \le i \le N-1} w_{m,i} \qquad (2.27)\end{aligned}$$

The ratio $\kappa_m = \frac{C_m}{\mu_m}$ is called the *condition number*.

Algorithmic Complexity and Approximate Solutions

The proposed projected gradient optimization method is summarized in Algorithm 2. Clearly, the computational complexity of one iteration is seen to be extremely low. Moreover, the shortcomings of the optimization method 1 presented in subsection 2.3.2 are dealt with:

- It is possible to solve the optimization problem inexactly by stopping the iterative optimization method before convergence to the exact solution $\mathbf{y}_m^*$ is reached. The iterates $\mathbf{y}_m^k$ of the proposed projected gradient method are feasible by construction. Moreover, the sequence $\{f(\mathbf{y}_m^k)\}$ can be proved to be monotonically decreasing. Hence, stopping the method after any number of iterations η will result in a feasible point $\mathbf{y}_m^\eta$ for which $f(\mathbf{y}_m^\eta) \le f(\mathbf{y}_m^0)$. We can then define the obtained solution accuracy as $\epsilon = f(\mathbf{y}_m^\eta) - f(\mathbf{y}_m^*)$.

- It is possible to derive polynomial upper and lower bounds on the *algorithmic complexity*, i.e. the number of necessary iterations of the optimization method as a function of the obtained solution accuracy ϵ, as we will show next.

For the class of convex optimization problems with strongly convex objective functions, a general lower bound on the algorithmic complexity was derived that holds for all iterative first-order methods [19, Ch. 2]:

Theorem 2.10 *For any starting point $\mathbf{y}_m^0 \in \mathbb{R}^N$, for any first-order projected gradient method, and for any closed convex set Q, there exists a strongly convex, continuously differentiable function f with Lipschitz constant C_m and convexity parameter $\mu_m = \frac{C_m}{\kappa_m}$ (where $\kappa_m > 1$), such that we have*

$$f(\mathbf{y}_m^k) - f(\mathbf{y}_m^*) \geq \frac{\mu_m}{2} \left(\frac{\sqrt{\kappa_m} - 1}{\sqrt{\kappa_m} + 1} \right)^{2k} ||\mathbf{y}_m^0 - \mathbf{y}_m^*||^2, \ \forall k \tag{2.28}$$

Also, for the same problem class, an upper bound on the algorithmic complexity of the projected gradient method used in Algorithm 2 was derived [19, Ch. 2]:

Theorem 2.11 *Let f be a strongly convex, continuously differentiable function with Lipschitz constant C_m and convexity parameter $\mu_m = \frac{C_m}{\kappa_m}$ (where $\kappa_m > 1$), and let Q be a convex feasible set. Then the projected gradient method described in Algorithm 2 with fixed stepsize $\frac{1}{C_m}$ generates a sequence $\{\mathbf{y}_m^k\}$ which converges as follows:*

$$f(\mathbf{y}_m^k) - f(\mathbf{y}_m^*) \leq \frac{\mu_m}{2} \left(\frac{\kappa_m - 1}{\kappa_m + 1} \right)^{2k} ||\mathbf{y}_m^0 - \mathbf{y}_m^*||^2, \ \forall k. \tag{2.29}$$

As the general algorithmic complexity lower bound (2.28) for first-order optimization methods, and the specific algorithmic complexity upper bound (2.29) for the projected gradient optimization method described in Algorithm 2 differ by an order of magnitude, it can be concluded that the proposed optimization method is not optimal in terms of convergence.

2.3.4 Optimization Method 3: Optimal Projected Gradient Descent

From the complexity bounds given in subsection 2.3.3 it might be expected that it is theoretically possible for a first-order optimization method to have a better convergence rate than optimization method 2. Indeed, if there exists a first-order method for which the complexity upper bound is proportional to the complexity lower bound for a given problem class, this method could be called *optimal* for that problem class [19]. In this subsection, we present a projected gradient optimization method that reaches an optimal convergence for the class of convex optimization problems with strongly convex objective functions. This method was first proposed in [19] and variants of the method

Algorithm 3 Optimal projected gradient method

```
Input $\mathbf{x}_m \in \mathbb{R}^N$, $\mathbf{y}_m^0 = \mathbf{c}_m^0 \in Q$, $\mathbf{W}_m$, $\gamma_m^0 \in (0,1)$, $L$, $U$
Output $\mathbf{y}_m^* \in \mathbb{R}^N$
Calculate Lipschitz constant $C_m$ [using (2.26)]
Calculate convexity parameter $\mu_m$ [using (2.27)]
$\kappa_m = \frac{C_m}{\mu_m}$
$k = 0$
while convergence is not reached do
    $\tilde{\mathbf{y}}_m^{k+1} = \mathbf{c}_m^k - \frac{1}{C_m}\nabla f(\mathbf{c}_m^k)$
    $\mathbf{y}_m^{k+1} = \Pi_Q(\tilde{\mathbf{y}}_m^{k+1})$    [using (2.23)]
    Calculate $\gamma_m^{k+1}$ from $(\gamma_m^{k+1})^2 = (1-\gamma_m^{k+1})(\gamma_m^k)^2 + \kappa_m\gamma_m^{k+1}$
    $\delta_m^k = \frac{\gamma_m^k(1-\gamma_m^k)}{(\gamma_m^k)^2+\gamma_m^{k+1}}$
    $\mathbf{c}_m^{k+1} = \mathbf{y}_m^{k+1} + \delta_m^k(\mathbf{y}_m^{k+1} - \mathbf{y}_m^k)$
    $k = k + 1$
end while
$\mathbf{y}_m^* = \mathbf{y}_m^k$
```

have been applied in diverse applications, e.g. for real-time model predictive control [20].

Algorithm 3 summarizes the optimal projected gradient optimization method. We note the following differences compared to Algorithm 2:

- Knowledge of the convexity parameter μ_m is incorporated.
- In each iteration, a standard projected gradient step is performed on a potentially infeasible weighted sum of two previous feasible iterates.

It is again possible to derive polynomial an upper bound on the algorithmic complexity of the projected gradient method used in Algorithm 3, i.e. the number of necessary iterations of this optimization method as a function of the solution accuracy ϵ [19, Ch. 2]:

Theorem 2.12 *Let f be a strongly convex, continuously differentiable function with Lipschitz constant C_m and convexity parameter $\mu_m = \frac{C_m}{\kappa_m}$ (where $\kappa_m > 1$), and let Q be a convex feasible set. Then the projected gradient method described in Algorithm 3 generates a sequence $\{\mathbf{y}_m^k\}$ which converges as follows:*

$$f(\mathbf{y}_m^k) - f(\mathbf{y}_m^*) \leq C_m \min\left\{\left(1 - \sqrt{\frac{\mu_m}{C_m}}\right)^k, \frac{4}{(k+2)^2}\right\} ||\mathbf{y}_m^0 - \mathbf{y}_m^*||^2. \quad (2.30)$$

From (2.28), the minimum number of necessary iterations to find $\mathbf{y}_m^k$ satisfying

Table 2.1: Algorithmic and arithmetic complexity of optimization methods 2 and 3

Algorithm	Algorithmic complexity	Arithmetic complexity per iteration
Method 2	$\kappa_m \log \frac{4C_m N U^2}{\epsilon}$	$\frac{51}{9} N \log_2 N - \frac{61}{18} N$
Method 3	$\sqrt{\kappa_m} \log \frac{4C_m N U^2}{\epsilon}$	$\frac{51}{9} N \log_2 N - \frac{7}{18} N$

$f(\mathbf{y}_m^k) - f(\mathbf{y}_m^*) \leq \epsilon$ is

$$k_m^{\min} = \frac{\sqrt{\kappa_m} - 1}{4} \left(\log \frac{1}{\epsilon} + \log \frac{\mu_m}{2} + 2 \log ||\mathbf{y}_m^0 - \mathbf{y}_m^*|| \right) \tag{2.31}$$

From (2.30), the maximum number of necessary iterations to find $\mathbf{y}_m^k$ satisfying $f(\mathbf{y}_m^k) - f(\mathbf{y}_m^*) \leq \epsilon$ is

$$k_m^{\max} = \sqrt{\kappa_m} \left(\log \frac{1}{\epsilon} + \log C_m + 2 \log ||\mathbf{y}_m^0 - \mathbf{y}_m^*|| \right) \tag{2.32}$$

Hence, the main term in the upper bound estimate (2.32), $\sqrt{\kappa_m} \log \frac{1}{\epsilon}$, is proportional to the lower bound (2.31), proving this optimization method to be an optimal first-order method for the class of strongly convex optimization problems.

Table 2.1 summarizes the computational complexity results of optimization methods 2 and 3. For both methods, the algorithmic complexity as well as the arithmetic complexity per iteration (in terms of number of real additions and multiplications) is given. The algorithmic complexity results are straightforwardly found from upper bounds (2.29) and (2.30) by incorporating $||\mathbf{y}_m^0 - \mathbf{y}_m^*|| = 2\sqrt{N}U$ as the (worst-case) maximum distance between two points in the feasible set Q defined in (2.21)[8], where it is assumed that $U = -L$. In the arithmetic complexity computations, estimates were used for the arithmetic complexity of a FFT with power-of-two length N, as derived in [24]: namely, $\frac{17}{9} N \log_2 N - \frac{89}{27} N + O(\log_2 N)$ for a real-data FFT, and $\frac{34}{9} N \log_2 N - \frac{124}{27} N + O(\log_2 N)$ for a complex-data FFT. In conclusion, optimization method 3 has a significantly better algorithmic complexity compared to optimization method 2, and this for a negligibly higher arithmetic complexity per iteration.

[8] The length of an N-agonal in an N-dimensional hypercube with side length $2U$ equals $2\sqrt{N}U$.

2.4 Simulation Results

2.4.1 Comparative Evaluation of Perceived Audio Quality

For audio quality evaluation purposes, a test database consisting of 24 audio excerpts was compiled (16 bit mono, 44.1 kHz). The excerpts were selected so as to cover different music styles, melodic and rythmic textures, instrumentations and dynamics, as well as different speech types (see Table 2.2 for details). A first set of excerpts (numbers 1-11) was extracted from different commercial audio CDs. A second set of excerpts (numbers 12-16) was extracted from an ITU CD-ROM associated to Recommendation BS.1387-1, which contains a database (DB3) used for validating the conformance of implementations to this recommendation [25]. A third set of excerpts (numbers 17-20) was extracted from the HINT speech database [26]. A fourth set of excerpts (numbers 21-24) was extracted from the VoxForge speech corpus [27].

Each audio signal in the test database was processed by three different clipping algorithms:

- *Hard symmetrical clipping*, where the input-output characteristic is defined as
$$y[n] = \begin{cases} x[n], & |x[n]| \leq U \\ \mathrm{sgn}(x[n])U, & |x[n]| > U \end{cases} \tag{2.33}$$
- *Soft symmetrical clipping* [3], where the input-output characteristic is defined as a linearized hyperbolic tangent function which is linear for inputs below a parametric amplitude level r,
$$y[n] = \begin{cases} x[n], & |x[n]| \leq r \\ \mathrm{sgn}(x[n]) \left[(U-r)\tanh\left(\frac{|x[n]|-r}{U-r}\right) + r\right], & |x[n]| > r \end{cases} \tag{2.34}$$
here used with parameter setting $r = 0.8U$.
- *Perception-based clipping* as described in this paper, with parameter values $N = 512$, $P = 128$, $\alpha = 0.04$, and application of optimization method 3 with a solution accuracy of $\epsilon = 10^{-12}$ for all instances of (2.2).

This processing was performed for eight clipping factors {0.70, 0.80, 0.85, 0.90, 0.95, 0.97, 0.98, 0.99}[9]. For each of a resulting total of 24×8×3=576 processed audio signals, two *objective measures of perceived audio quality* were calculated, which aim to predict the subjective audio quality score that would be attributed to the processed audio signal by an average human listener. Taking

[9] Note that the clipping factor above which a normal-hearing listener does not perceive hard clipping distortion has been subjectively evaluated to be higher than 0.99 for speech [39], and 0.997 for music [40].

Table 2.2: Audio excerpts database used for comparative audio quality evaluation of clipping algorithms

Nr.	Name	Texture	Composition	Style	Duration [s]	$\text{Sample}_{\text{start}}$	$\text{Sample}_{\text{end}}$	Origin
1	poulenc.wav	polyphonic	instrumental	classical	17.8	400000	1183000	[28]
2	rhcp.wav	polyphonic	instrumental	rock	9.8	468996	900000	[29]
3	pierle.wav	polyphonic	instrumental+vocal	pop	11.7	2234000	2750000	[30]
4	mascagni.wav	polyphonic	instrumental+vocal	classical	16.1	1	711000	[31]
5	chopin.wav	monophonic	instrumental	classical	17.8	50000	836200	[32]
6	kraftwerk.wav	polyphonic	instrumental	electronic	17.2	7480000	8240000	[33]
7	baloji.wav	polyphonic	instrumental+vocal	hip hop	14.4	8084000	8719000	[34]
8	arcade.wav	polyphonic	instrumental+vocal	pop	13.6	1	600000	[35]
9	strokes.wav	polyphonic	instrumental	rock	13.6	1	600000	[36]
10	beethoven.wav	monophonic	instrumental	classical	15.4	1580000	2260000	[37]
11	albeniz.wav	monophonic	instrumental	classical	11.8	7800000	8320000	[38]
12	breftri.wav	monophonic	instrumental	classical	19.7	1	869675	[25]
13	crefsax.wav	monophonic	instrumental	classical	10.9	1	479026	[25]
14	freftr1.wav	monophonic	instrumental	classical	13.1	1	577887	[25]
15	grefcla.wav	monophonic	instrumental	classical	6.9	1	302534	[25]
16	mrefcla.wav	monophonic	instrumental	classical	7.4	1	326500	[25]
17-20	hint{01-04}.wav	monophonic	speech	male	4 × 17.0	1	749700	[26]
21-24	silvia{01-04}.wav	monophonic	speech	female	4 × 8.0	1	352800	[27]

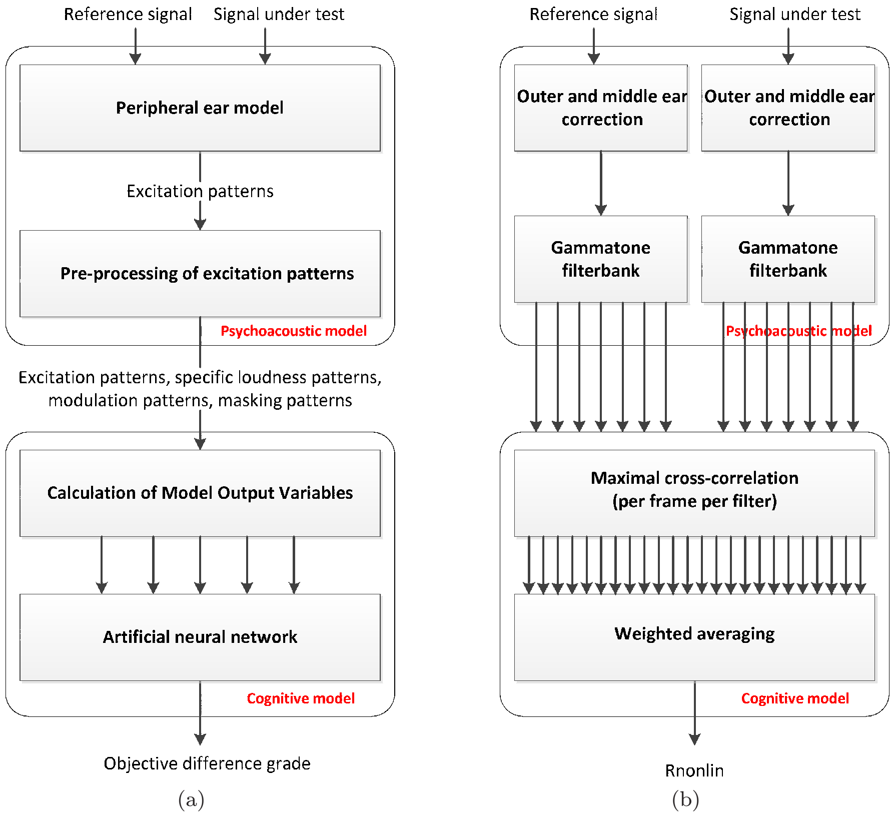

Reference signal
Signal under test
Peripheral ear model
Excitation patterns
Pre-processing of excitation patterns
Psychoacoustic model
Excitation patterns, specific loudness patterns, modulation patterns, masking patterns
Calculation of Model Output Variables
Artificial neural network
Cognitive model
Objective difference grade
(a)
Reference signal
Signal under test
Outer and middle ear correction
Outer and middle ear correction
Gammatone filterbank
Gammatone filterbank
Psychoacoustic model
Maximal cross-correlation (per frame per filter)
Weighted averaging
Cognitive model
Rnonlin
(b)

Figure 2.3: Block diagrams of the used objective measures of perceived audio quality : (a) PEAQ Basic version (adapted from [25]) (b) Rnonlin

a reference signal (i.e. the clean signal) and a signal under test (i.e. the processed signal) as an input, such an objective measure of perceived audio quality is calculated through sequential application of a psychoacoustic model and a cognitive model, and resultingly attributes a perceived audio quality score to the signal under test with respect to the reference signal (see Figure 2.3).

A first objective measure of perceived audio quality was calculated using the Basic Version of the PEAQ (*Perceptual Evaluation of Audio Quality*) recommendation [25]. A block diagram representation of this method is shown in Figure 2.3(a). The resulting Objective Difference Grade (ODG) predicts the basic audio quality of the signal under test with respect to the reference signal, and has a range between 0 and - 4, corresponding to the ITU-R five grade impairment scale depicted in Figure 2.4.

Table 2.3: Objective audio quality scores for hard clipping (H), soft clipping (S) and perception-based clipping (P-B) for two selected clipping factors {0.99,0.90}: PEAQ ODG and Rnonlin. Highest scores per audio excerpt are highlighted.

Nr.	PEAQ						Rnonlin					
	CF=0.99			CF=0.90			CF=0.99			CF=0.90		
	H	S	P-B	H	S	P-B	H	S	P-B	H	S	P-B
1	-1.86	-1.79	-1.13	-3.85	-3.84	-3.58	99.14	99.15	99.57	87.72	87.87	93.22
2	-0.38	-0.31	-0.36	-0.93	-0.94	-0.84	99.29	99.38	99.56	92.68	93.54	95.99
3	0.13	0.11	-0.17	-1.26	-0.74	-1.34	99.93	99.87	99.97	95.35	95.88	97.61
4	-2.98	-2.10	-2.03	-3.86	-3.84	-3.77	98.62	98.77	99.41	88.60	89.68	93.37
5	-3.09	-2.21	-1.08	-3.89	-3.88	-3.71	96.97	97.01	98.94	79.89	81.52	89.46
6	-0.62	-0.52	-0.16	-2.55	-2.31	-1.52	99.12	99.22	99.71	91.29	91.76	95.94
7	-0.42	-0.39	-0.14	-1.89	-1.64	-1.07	98.53	98.63	99.55	90.64	91.18	94.73
8	-0.24	-0.21	-0.14	-1.77	-1.28	-2.00	99.41	99.46	99.70	93.36	94.14	95.45
9	-0.51	-0.44	-0.40	-3.75	-3.68	-3.51	98.93	99.03	99.52	88.36	88.99	93.46
10	-3.71	-1.65	-1.71	-3.82	-3.82	-3.41	98.32	98.99	99.51	90.02	91.58	94.79
11	-1.65	-0.67	-0.59	-3.70	-3.44	-1.82	97.63	98.22	99.26	85.43	87.10	92.34
12	-1.92	-1.93	-1.42	-3.80	-3.80	-3.69	98.67	98.66	99.50	90.68	90.68	94.07
13	-0.67	-0.46	-0.45	-3.21	-2.47	-2.25	97.86	98.31	99.44	85.07	86.08	91.13
14	-3.45	-3.46	-0.88	-3.85	-3.85	-3.60	96.35	96.25	99.54	87.84	87.75	94.37
15	-1.41	-1.37	-0.95	-3.56	-3.25	-3.17	97.41	97.40	99.02	87.35	88.21	93.16
16	-1.59	-1.50	-1.07	-3.64	-3.40	-3.35	97.46	97.46	99.03	87.62	88.32	93.07
17	-3.86	-3.78	-3.06	-3.91	-3.91	-3.83	95.47	95.93	98.14	82.61	83.99	88.90
18	-3.84	-3.73	-2.67	-3.91	-3.91	-3.85	96.54	96.93	98.49	82.49	83.84	88.94
19	-3.80	-3.51	-2.81	-3.90	-3.89	-3.78	96.12	96.73	98.42	82.42	84.24	89.14
20	-3.80	-3.61	-2.38	-3.90	-3.90	-3.81	96.61	97.08	98.55	84.42	85.78	90.25
21	-1.07	-1.10	-0.55	-3.57	-3.55	-2.16	98.31	98.37	99.24	88.26	88.77	92.37
22	-1.00	-0.98	-0.56	-3.42	-3.36	-1.99	98.06	98.22	99.12	88.88	89.43	92.64
23	-0.80	-0.76	-0.55	-3.40	-3.37	-2.02	97.91	98.07	99.19	88.38	88.99	92.85
24	-0.62	-0.60	-0.47	-3.18	-3.14	-1.63	98.25	98.34	99.27	87.86	88.52	92.44
Avg.	**-1.80**	**-1.54**	**-1.07**	**-3.27**	**-3.13**	**-2.74**	**97.95**	**98.14**	**99.24**	**87.80**	**88.66**	**92.90**

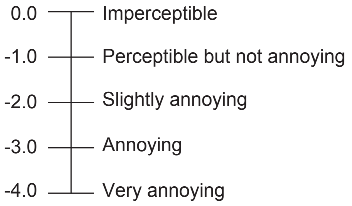


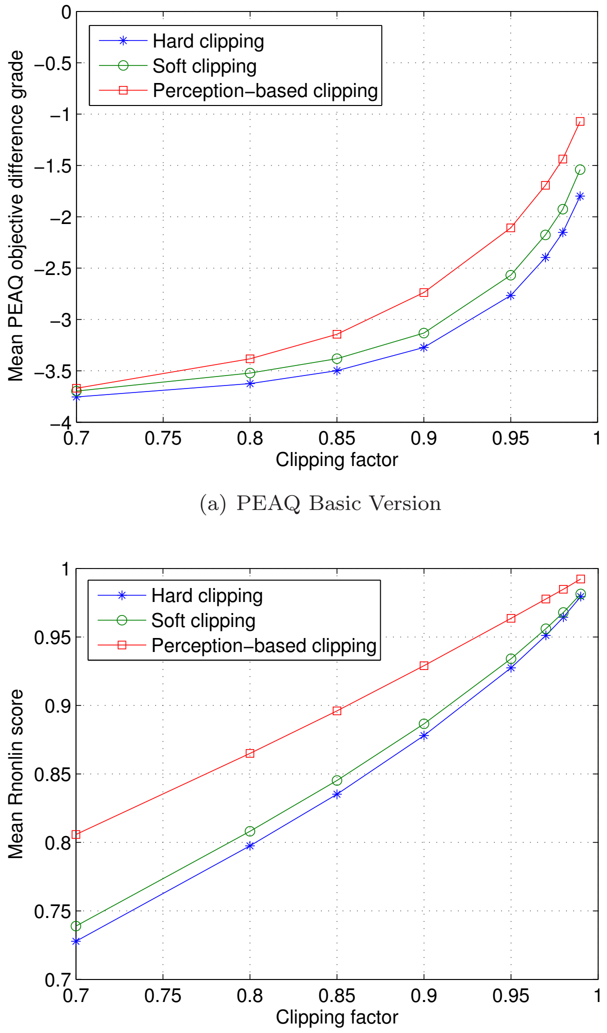
Figure 2.4: The ITU-R five-grade impairment scale

Table 2.4: P-values from one-tailed paired t-tests on audio quality scores. Significant P-values with respect to $\alpha_{TT} = 0.05$ in bold.

H_0	$E[S_H] \geq E[S_{PB}]$		$E[S_S] \geq E[S_{PB}]$	
	PEAQ	Rnonlin	PEAQ	Rnonlin
CF=0.70	0.0766	**<0.0001**	0.3566	**<0.0001**
CF=0.80	**0.0042**	**<0.0001**	0.1106	**<0.0001**
CF=0.85	**0.0017**	**<0.0001**	**0.0398**	**<0.0001**
CF=0.90	**<0.0001**	**<0.0001**	**0.0026**	**<0.0001**
CF=0.95	**<0.0001**	**<0.0001**	**<0.0001**	**<0.0001**
CF=0.97	**<0.0001**	**<0.0001**	**<0.0001**	**<0.0001**
CF=0.98	**<0.0001**	**<0.0001**	**<0.0001**	**<0.0001**
CF=0.99	**<0.0001**	**<0.0001**	**0.0004**	**<0.0001**

A second objective measure used here was specifically designed to predict the perceived audio quality of nonlinearly distorted signals and is described in [41]. A block diagram representation of this method is shown in Figure 2.3(b). The resulting *Rnonlin* score is a perceptually relevant measure of distortion. Rnonlin values have a range between 0 and 1 and are seen to decrease for increasing perceptible distortion (i.e. with decreasing audio quality).

The results of these simulations are shown in Figure 2.5. In Figure 2.5(a), the average PEAQ ODG score over all 24 audio signals is plotted as a function of the clipping factor, and this for the three different clipping techniques. Analogously, Figure 2.5(b) shows the results for the Rnonlin measure. The obtained results for both audio quality measures are seen to be in accordance with each other. Logically, we observe a monotonically increasing average audio quality score for increasing clipping factors. Soft clipping is seen to result in slightly higher average objective audio quality scores than hard clipping. Clearly, the perception-based clipping technique is seen to result in significantly higher average objective audio quality scores than the other clipping techniques, and this for all considered clipping factors. In Table 2.3, the full simulation results per individual audio excerpt are provided for two selected clipping factors (0.99 and 0.90). For each audio excerpt, the highest score for each objective measure is highlighted.



(a) PEAQ Basic Version

(b) Rnonlin

Figure 2.5: Comparative evaluation of different clipping techniques in terms of objective perceived audio quality: (a) mean PEAQ ODG and (b) mean Rnonlin scores for signals processed by hard clipping, soft clipping and perception-based clipping as a function of the clipping factor

In order to infer statistically significant conclusions on the comparative audio quality performance of the three clipping algorithms under study, a statistical analysis was performed on the obtained set of PEAQ ODG and Rnonlin scores. Let us represent the audio quality scores resulting from hard clipping, soft-clipping and perception-based clipping for a given clipping factor by random variables S_H, S_S, and S_{PB}, respectively. Under the assumption that these random variables follow a normal probability distribution[10], we tested the two following statistical hypotheses based on the sample data. The first null hypothesis H_0 and its alternative H_1 are formulated as follows,

$$H_0 \ : \ E[S_H] \geq E[S_{PB}] \tag{2.35}$$

$$H_1 \ : \ E[S_{PB}] > E[S_H] \tag{2.36}$$

The second null hypothesis H_0 and its alternative H_1 are formulated as follows,

$$H_0 \ : \ E[S_S] \geq E[S_{PB}] \tag{2.37}$$

$$H_1 \ : \ E[S_{PB}] > E[S_S] \tag{2.38}$$

These two statistical hypotheses were tested for all considered clipping factors, and for both audio quality measures. All statistical hypotheses were tested using one-tailed paired t-tests with significance level $\alpha_{TT} = 0.05$. The resulting one-sided P-values are synthesized in Table 2.4. For PEAQ scores, the first null hypothesis (2.35) can be rejected in favor of the alternative (2.36) for clipping factors of 0.80 and higher. The second null hypothesis (2.37) can be rejected in favor of the alternative (2.38) for clipping factors of 0.85 and higher. For Rnonlin scores, both null hypotheses can be rejected in favor of the alternative for all considered clipping factors. We can conclude that there is strong statistical evidence that the perception-based clipping technique will in general deliver signals with a higher perceptual audio quality compared to the other considered clipping techniques, and this for moderate to high clipping factors.

2.4.2 Experimental Evaluation of Algorithmic Complexity

In order to assess experimentally the validity of the theoretical algorithmic complexity bounds of the projected gradient optimization methods 2 and 3 described in subsections 2.3.3 and 2.3.4, a simulation was conducted as follows. For optimization methods 2 and 3, the number of iterations needed to reach solution accuracies $\epsilon = \{10^{-4}, 10^{-5}, ..., 10^{-10}\}$ was determined for a subset of the instances of optimization problem (2.2) occurring in our test database of 24 audio signals. This was performed for six clipping factors {0.85, 0.90, 0.95, 0.97, 0.98, 0.99}.

[10] The validity of this assumption was verified for our sample data using the Jarque-Bera normality test [42] at significance level $\alpha_{JB} = 0.05$.

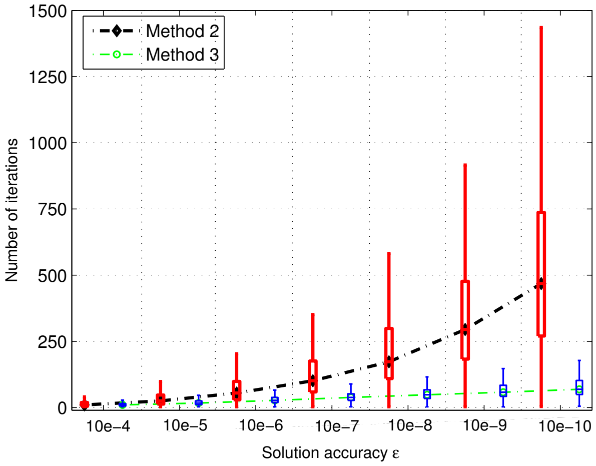
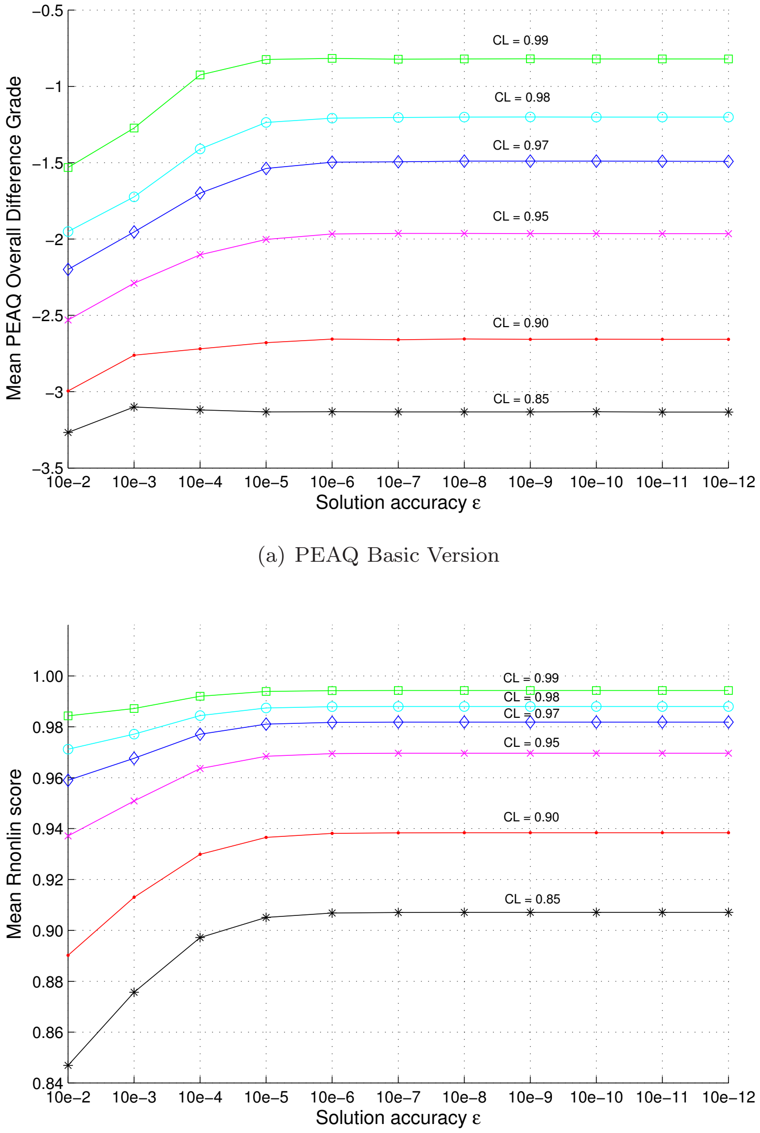


Figure 2.6: Boxplot of number of iterations vs solution accuracy for optimization methods 2 and 3

In Figure 2.6, the simulation results per optimization method are summarized in the form of a boxplot for each solution accuracy, depicting graphically the minimum, lower quartile, median, upper quartile and maximum values of the number of iterations. The dotted lines connect the median number of iterations of both optimization methods for different solution accuracies. We observe that the median (the same holds for the maximum) number of iterations follows a different curve depending on the optimization method: optimization method 3 is seen to have a significantly better algorithmic complexity compared to optimization method 2, as was derived theoretically in subsections 2.3.3 and 2.3.4.

2.4.3 Applicability in Real-Time Context: Effect of Solution Accuracy ϵ on Perceived Audio Quality

In a real-time processing context, the number of clock cycles that can be spent on solving an instance of optimization problem (2.2) is strictly limited. In view of this, several computationally efficient convex optimization methods tailored to the optimization problem were presented in Section 2.3. It was shown that the iterative projected gradient optimization methods 2 and 3 have the advantage that approximate solutions can be computed, which makes it possible to adhere to the imposed real-time deadlines. The question remains as to how approximately solving the optimization problems affects the perceived audio quality of the resulting output signal. In order to assess this effect,



(a) PEAQ Basic Version

(b) Rnonlin

Figure 2.7: Mean objective audio quality scores for different solution accuracies for perception-based clipping (a) PEAQ ODG (b) Rnonlin

PEAQ ODG and Rnonlin scores were calculated for a subset of the signals in our test database, each of which was processed by perception-based clipping for optimization problem solution accuracies $\epsilon = \{10^{-2}, 10^{-3}, ..., 10^{-12}\}$. This was performed for six clipping factors $\{0.85, 0.90, 0.95, 0.97, 0.98, 0.99\}$.

In Figures 2.7(a) and 2.7(b), the resulting mean PEAQ ODG and Rnonlin scores over all audio signals are plotted as a function of the solution accuracy, and this for all considered clipping factors. We observe that, according to both measures, the mean audio quality is affected negatively for low solution accuracies, increases nearly monotonically with increasing solution accuracies, and saturates at a solution accuracy that depends on the clipping factor. Irrespective of the clipping factor, we observe that no further improvement in mean audio quality scores is obtained for higher solution accuracies than 10^{-6}. Hence, $\epsilon = 10^{-6}$ can be put forward as an experimentally established sufficient solution accuracy for all considered clipping factors, such that no sacrifice in terms of audio quality is made.

For parameter values $N = 512$, $P = 128$ and sampling rate of 44.1 kHz, the real-time computation time limit for solving one instance of optimization problem (2.2) is equal to 8.7 ms. In our simulation setting[11], this corresponds to an iteration limit of roughly 200 iterations for optimization methods 2 or 3 (neglecting here the small difference in arithmetic complexity per iteration between both methods as shown in Table 2.1). Looking back at Figure 2.6, we see that even for the worst-case instances in our test database, optimization method 3 meets the real-time iteration limit for solution accuracies up to 10^{-10}, which largely surpasses the required solution accuracy of 10^{-6}.

2.5 Conclusions

In this paper, we have presented a novel algorithm for real-time perception-based clipping of audio signals. By including a psychoacoustic model and embedding convex optimization into the algorithm, it is possible to explicitly minimize the perceptible distortion introduced by clipping. From comparative audio quality evaluation experiments, it has been concluded that the perception-based clipping algorithm results in significantly higher objective audio quality scores than standard clipping techniques, and this for moderate to high clipping factors. Furthermore, three optimization methods aimed at efficiently solving the convex optimization problems were derived. The reliable use of optimization method 3 in real-time applications was seen to be supported by theoretically derived complexity bounds as well as by simulation experiments.

In a broader view, the results presented in this paper suggest that embedded

[11] All simulations were performed on a GenuineIntel CPU @2826 Mhz

convex optimization is a very promising paradigm in real-time audio processing applications, with numerous potential applications, of which we point out e.g. speech enhancement and acoustic echo cancellation.

Bibliography

[1] G. A. Constantinides, P. Y. K. Cheung, and W. Luk, “Synthesis of saturation arithmetic architectures,” *ACM Trans. on Design Automation of Electron. Syst.*, vol. 8, no. 3, pp. 334–354, Jul. 2003.

[2] U. Zölzer et al., *DAFX:Digital Audio Effects*, U. Zölzer, Ed. John Wiley & Sons, May 2002.

[3] A. N. Birkett and R. A. Goubran, “Nonlinear loudspeaker compensation for hands free acoustic echo cancellation,” *Electron. Lett.*, vol. 32, no. 12, pp. 1063–1064, Jun. 1996.

[4] F. Foti, “Aliasing distortion in digital dynamics processing, the cause, effect, and method for measuring it: The story of ’digital grunge!’,” in *Preprints AES 106th Conv.*, Munich, Germany, May 1999, Preprint no. 4971.

[5] C.-T. Tan, B. C. J. Moore, and N. Zacharov, “The effect of nonlinear distortion on the perceived quality of music and speech signals,” *J. Audio Eng. Soc.*, vol. 51, no. 11, pp. 1012–1031, Nov. 2003.

[6] C.-T. Tan and B. C. J. Moore, “Perception of nonlinear distortion by hearing-impaired people,” *Int. J. Audiol.*, vol. 47, pp. 246–256, May 2008.

[7] A. Adler, V. Emiya, M. Jafari, M. Elad, R. Gribonval, and M. D. Plumbley, “A constrained matching pursuit approach to audio declipping,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., Prague, Czech Republic*, May 2011, pp. 329–332.

[8] J. Han, G. J. Mysore, and B. Pardo, “Audio imputation using the Non-negative Hidden Markov Model,” in *Lecture Notes in Computer Science: Latent Variable Analysis and Signal Separation*, 2012.

[9] T. Painter and A. Spanias, “Perceptual coding of digital audio,” *Proc. IEEE*, vol. 88, no. 4, pp. 451–515, Apr. 2000.

[10] D. De Koning and W. Verhelst, “On psychoacoustic noise shaping for audio requantization,” in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Proc.*, Hong Kong, Apr. 2003, pp. 413–416.

[11] J. Mattingley and S. Boyd, “Real-time convex optimization in signal processing,” *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 50–61, 2010.

[12] B. Defraene, T. van Waterschoot, H. J. Ferreau, M. Diehl, and M. Moonen, “Perception-based clipping of audio signals,” in *2010 European Signal Processing Conference (EUSIPCO-2010)*, Aalborg, Denmark, Aug. 2010, pp. 517–521.

[13] B. Defraene, T. van Waterschoot, M. Diehl, and M. Moonen, “A fast projected gradient optimization method for real-time perception-based clipping of audio signals,” in *Proc. ICASSP 2011*, Prague, Czech Republic, May 2011, pp. 333–336.

[14] ISO/IEC, “11172-3 Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s - Part 3: Audio,” 1993.

[15] E. Terhardt, “Calculating virtual pitch,” *Hearing Res.*, vol. 1, no. 2, pp. 155 – 182, 1979.

[16] E. Zwicker and H. Fastl, *Psychoacoustics: facts and models*, 2nd ed. Springer, 1999.

[17] P. Davis, *Circulant matrices.* Wiley, New York, 1979.

[18] R. M. Gray, *Toeplitz and circulant matrices: a review.* Stanford University, 2001.

[19] Y. Nesterov, *Introductory lectures on convex optimization.* Springer, 2004.

[20] S. Richter, C. Jones, and M. Morari, “Real-Time Input-Constrained MPC Using Fast Gradient Methods,” in *Conference on Decision and Control (CDC)*, Shanghai, China, Dec. 2009, pp. 7387 – 7393.

[21] H. J.Ferreau, H. G. Bock, and M. Diehl, “An online active set strategy to overcome the limitations of explicit MPC,” *Int. J. Robust Nonlinear Contr.*, Jul. 2008.

[22] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge University Press, 2004.

[23] D. P. Bertsekas, *Nonlinear programming, 2nd ed.* Belmont, Massachusetts: Athena Scientific, 1999.

[24] S. G. Johnson and M. Frigo, “A modified split-radix FFT with fewer arithmetic operations,” *IEEE Trans. Signal Processing*, vol. 55, no. 1, pp. 111–119, 2007.

[25] International Telecommunications Union Recommendation BS.1387, “Method for objective measurements of perceived audio quality,” 1998.

[26] M. Nilsson, S. D. Soli, and J. A. Sullivan, "Development of the hearing in noise test for the measurement of speech reception thresholds in quiet and in noise," *Journal of the Acoustical Society of America*, vol. 95, no. 2, pp. 1085–1099, 1994.

[27] VoxForge, "Free speech corpus," http://www.voxforge.org, Jan. 2012.

[28] F. Poulenc, "Sonata for flute and piano (Cantilena)," French Flute Music, Naxos 8.557328, 2005.

[29] Red Hot Chili Peppers, "Californication," Californication, Warner Bros. Records 9362473862, 1999.

[30] An Pierlé & White Velvet, "Mexico," An Pierlé & White Velvet, PIAS Recordings 941.0170.020, 2006.

[31] P. Mascagni, "Cavalleria rusticana," 2013. [Online]. Available: ftp://ftp.esat.kuleuven.be/pub/SISTA/bdefraen/reports/mascagni.wav

[32] F. Chopin, "Waltz op. 69 no. 2," Favourite Piano Works (Vladimir Ashkenazy), Decca 02894448302, 1995.

[33] Kraftwerk, "Tour de France Etape 1," Minimum-Maximum, EMI Music 724356061620, 2005.

[34] Baloji, "Entre les lignes," HUMO's Top 2007, EMI Music 5099950986125, 2007.

[35] A. Fire, "Deep blue," The Suburbs, Mercury 602527426297, 2010.

[36] T. Strokes, "Last nite," Is this it, RCA 07836804521, 2001.

[37] L. van Beethoven, "Piano sonata op. 13," Beethoven Favourite Piano Sonatas, Deutsche Grammophon 028947797586, 2011.

[38] I. Albeniz, "Iberia, book i," Lang Lang Live in Vienna, Sony Classical 886977190025, 2010.

[39] J. Kates and L. Kozma-Spytek, "Quality ratings for frequency-shqped peak-clipped speech," *Journal of the Acoustical Society of America*, vol. 95, no. 6, pp. 3586–3594, 1994.

[40] J. Moir, "Just detectable distortion levels," *Wireless World*, vol. 87, no. 1541, 1981.

[41] C.-T. Tan, B. C. J. Moore, N. Zacharov, and V.-V. Mattila, "Predicting the perceived quality of nonlinearly distorted music and speech signals," *J. Audio Eng. Soc.*, vol. 52, no. 7–8, pp. 699–711, Jul. 2004.

[42] C. M. Jarque and A. K. Bera, "Efficient tests for normality, homoscedasticity and serial independence of regression residuals," *Economic Letters*, vol. 6, no. 3, pp. 255–259, 1980.

Chapter 3

Loudspeaker Precompensation

Embedded-Optimization-Based Loudspeaker Precompensation Using a Hammerstein Loudspeaker Model

Bruno Defraene, Toon van Waterschoot, Moritz Diehl,
and Marc Moonen

Submitted to *IEEE Trans. Audio Speech Language Process.*,
Nov. 2013.

Contributions of first author

- literature study
- co-design of loudspeaker precompensation problem formulation
- co-analysis of relation between memoryless nonlinearity and problem structure
- co-development of gradient optimization method
- co-development of projected gradient optimization method
- co-derivation of algorithmic complexity bounds
- co-design of evaluation experiments
- software implementation and computer simulations
- co-identification of Hammerstein loudspeaker model
- co-interpretation of simulation results
- co-formulation of conclusions
- text redaction and editing

Abstract

This paper presents an embedded-optimization-based loudspeaker precompensation algorithm using a Hammerstein loudspeaker model, i.e. a cascade of a memoryless nonlinearity and a linear finite impulse response filter. The loudspeaker precompensation consists in a per-frame signal optimization. In order to minimize the perceptible distortion incurred in the loudspeaker, a psychoacoustically motivated optimization criterion is proposed. The resulting per-frame signal optimization problems are solved efficiently using first-order optimization methods. Depending on the invertibility and the smoothness of the memoryless nonlinearity, different first-order optimization methods are proposed and their convergence properties are analyzed. Objective evaluation experiments using synthetic and identified loudspeaker models show that the proposed loudspeaker precompensation algorithm provides a significant audio quality improvement, especially so at high playback levels.

3.1 Introduction

Achieving a high perceived audio quality is a main concern in the development of any audio reproduction system. In general, loudspeakers have a non-ideal response introducing both linear and nonlinear distortion in the reproduced audio signal. Linear distortion involves changes in the relative amplitudes and phases of the frequency components constituting the original audio signal. Nonlinear distortion involves the introduction of frequency components that are not present in the original audio signal, and is a notably prominent problem in small and low-cost loudspeakers, which are ubiquitous in mobile devices, especially so at high playback levels [1].

The presence of linear and nonlinear distortion has been found to result in a significant degradation of the perceived audio quality, both when present separately [2] and simultaneously [3]. Linear distortion is typically perceived as affecting timbre or tone quality, whereas nonlinear distortion is typically perceived as adding harshness or noisiness, or as the perception of sounds that are not present in the original signal, such as crackles or clicks.

Loudspeaker precompensation techniques aim at reducing the effects caused by the non-ideal loudspeaker characteristics. The idea is to apply a digital precompensation operation in cascade with the audio reproduction channel to counteract the loudspeaker response errors and nonlinearities. Traditionally, loudspeakers have been modeled by *linear* systems such as FIR filters, IIR filters, warped filters or Kautz filters. The aim of linear loudspeaker precompensation (also known as equalization) techniques is then to identify/approximate and apply an inverse digital filtering to the audio signal prior to playback

[4]. Nonlinear behaviour can be taken into account by using nonlinear loudspeaker models such as Hammerstein models, Wiener-Hammerstein models and Volterra models. The aim of nonlinear loudspeaker precompensation techniques is then to invert the nonlinear system under consideration [5].

This paper presents an embedded-optimization-based precompensation algorithm for linear as well as nonlinear distortions incurred in loudspeakers, using a Hammerstein loudspeaker model. The loudspeaker precompensation consists in a per-frame signal optimization. In order to minimize the perceptible distortion incurred in the loudspeaker, a psychoacoustic model is incorporated which captures knowledge about the human perception of sound.

This paper builds on the embedded-optimization-based loudspeaker precompensation algorithms recently proposed in [6]-[7], of which the main ideas will be reviewed in this paper, thereby introducing three major novel contributions:

1. The psychoacoustic principles and psychoacoustic model applied for perceptible distortion minimization are elaborated in detail.
2. Different gradient optimization methods are proposed for efficiently solving the per-frame signal optimization problems and their convergence properties are analyzed. Each optimization method is tailored to a given class of memoryless nonlinearities.
3. Extensive audio quality evaluation experiments are performed and reported, both using synthetic Hammerstein loudspeaker models (comprising different classes of memoryless nonlinearities), and using identified Hammerstein loudspeaker models.

This paper is organized as follows. In Section 3.2, the Hammerstein loudspeaker model is introduced and an embedded-optimization-based precompensation algorithm is proposed. In Section 3.3, three optimization methods for efficiently solving the per-frame signal optimization problems are proposed, where each optimization method is designed for a given class of memoryless nonlinearities. In Section 3.4, the proposed loudspeaker precompensation algorithm is evaluated using synthetic and identified Hammerstein loudspeaker models. In Section 3.5, some concluding remarks are presented.

3.2 Embedded-Optimization-Based Precompensation

3.2.1 Hammerstein Model Description

The loudspeaker is modeled by a Hammerstein model, i.e. a cascade of a memoryless nonlinearity and a linear finite impulse response (FIR) filter. The FIR filter has an impulse response $h[n], n = 0...L$. The memoryless nonlinearity

$g(x)$ is represented as a linear combination of P basis functions,

$$g(x) = \sum_{p=1}^{P} c_p \psi_p(x) = \boldsymbol{\psi}^T(x)\mathbf{c} \tag{3.1}$$

where the basis functions are stacked in a vector $\boldsymbol{\psi}(x) = [\psi_1(x), ..., \psi_P(x)]^T$ and the corresponding coefficients are stacked in a vector $\mathbf{c} = [c_1, ..., c_P]^T$.

A per-frame processing of the digital input audio signal $x[n]$ will be applied, employing input frames $\mathbf{x}_m = [x_{m,1}, ..., x_{m,N}]^T \in \mathbb{R}^N, m = 0, 1...M$, with $N \geq L$, and $x_{m,i} = x[mN + i]$. The output $\mathbf{g}(\mathbf{x}_m)$ of the memoryless nonlinearity for a given input frame $\mathbf{x}_m$ is straightforwardly constructed using the relation (3.1),

$$\begin{aligned} \mathbf{g}(\mathbf{x}_m) &= [g(x_{m,1}), ..., g(x_{m,N})]^T \\ &= \boldsymbol{\Psi}(\mathbf{x}_m)\mathbf{c} \end{aligned} \tag{3.2}$$

where the basis function vectors for the different samples are assembled in a matrix $\boldsymbol{\Psi}(\mathbf{x}_m) = [\boldsymbol{\psi}(x_{m,1}), ..., \boldsymbol{\psi}(x_{m,N})]^T$.

The output frame $\mathbf{y}_m$ of the Hammerstein model can then be written as

$$\mathbf{y}_m = \mathbf{H}_m \mathbf{g}(\mathbf{x}_m) + \tilde{\mathbf{H}}_m \mathbf{g}(\mathbf{x}_{m-1}) \tag{3.3}$$

where the matrices $\mathbf{H}_m \in \mathbb{R}^{N \times N}$ and $\tilde{\mathbf{H}}_m \in \mathbb{R}^{N \times N}$ implement a convolution operation with the FIR filter $h[n]$ as follows,

$$\mathbf{H}_m = \begin{bmatrix} h[0] & 0 & \dots & \dots & \dots & \dots & 0 \\ h[1] & h[0] & 0 & \dots & \dots & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & & & \vdots \\ h[L] & & \ddots & \ddots & \ddots & & \vdots \\ 0 & \ddots & & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & & \ddots & \ddots & 0 \\ 0 & \dots & 0 & h[L] & \dots & h[1] & h[0] \end{bmatrix} \tag{3.4}$$

$$\tilde{\mathbf{H}}_m = \begin{bmatrix} 0 & \dots & 0 & h[L] & \dots & h[2] & h[1] \\ 0 & \dots & 0 & 0 & h[L] & \dots & h[2] \\ \vdots & & & & \ddots & \ddots & \vdots \\ \vdots & & & & & \ddots & h[L] \\ \vdots & & & & & & 0 \\ \vdots & & & & & & \vdots \\ 0 & 0 & \dots & \dots & \dots & \dots & 0 \end{bmatrix}. \tag{3.5}$$

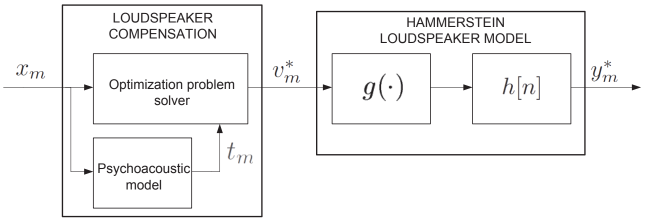


Figure 3.1: Embedded-optimization-based precompensation: schematic overview.

3.2.2 Embedded-Optimization-Based Precompensation

Figure 3.1 shows the operation of the proposed embedded-optimization-based loudspeaker precompensation technique. Before it is fed into the loudspeaker, the input frame $\mathbf{x}_m$ passes through the loudspeaker precompensation block. For a given input frame $\mathbf{x}_m$, the loudspeaker precompensation consists of the following steps:

1. Calculate the global masking threshold $\mathbf{t}_m \in \mathbb{R}^{\frac{N}{2}+1}$ of the input frame $\mathbf{x}_m$ using a psychoacoustic model (see Subsection 3.2.3).
2. Calculate a precompensated input frame $\mathbf{v}_m^* \in \mathbb{R}^N$ as the solution of an optimization problem, such that the corresponding output frame $\mathbf{y}_m^*$ is perceptually as close as possible to $\mathbf{x}_m$.

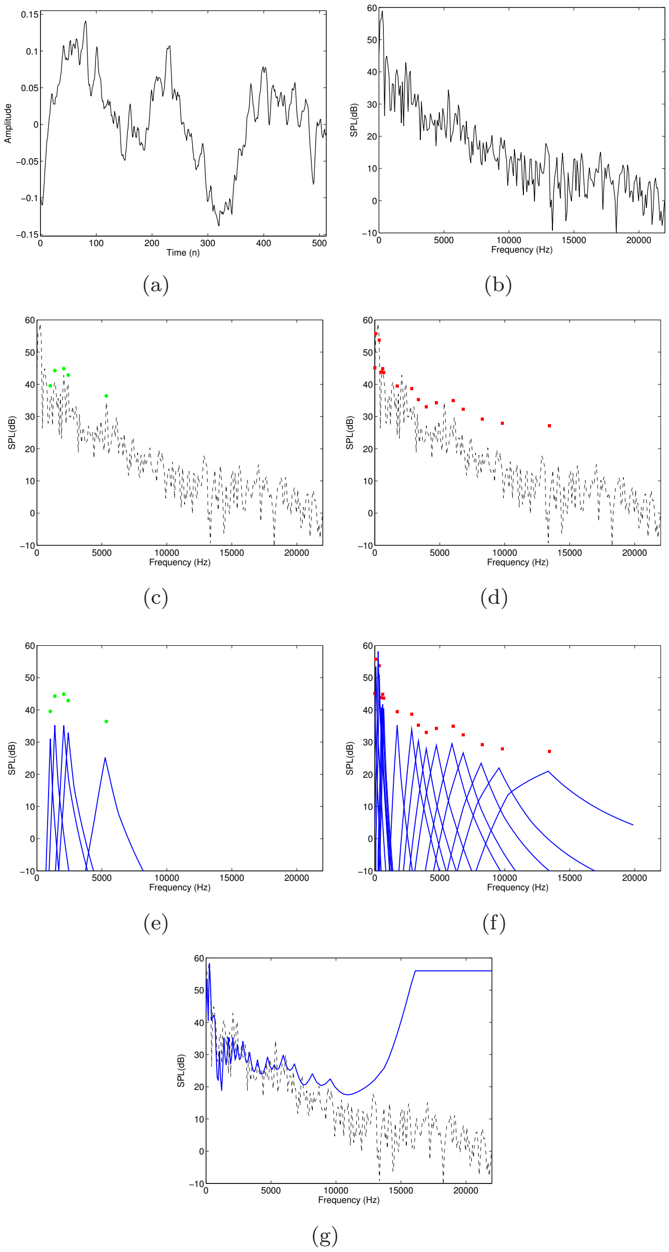
The precompensated input frame $\mathbf{v}_m^*$ is calculated from the knowledge of the input frame $\mathbf{x}_m$ and its masking threshold $\mathbf{t}_m$. The objective function reflects the amount of perceptible distortion added between $\mathbf{y}_m$ and $\mathbf{x}_m$,

$$\mathbf{v}_m^* = \underset{\mathbf{v}_m \in \mathbb{R}^N}{\arg\min} \ \frac{1}{2N} \sum_{i=0}^{N-1} w_{m,i} |Y_m(e^{j\omega_i}) - X_m(e^{j\omega_i})|^2 \tag{3.6}$$

where $\omega_i = (2\pi i)/N$ represents the discrete frequency variable, $X_m(e^{j\omega_i})$ and $Y_m(e^{j\omega_i})$ are the discrete frequency components of $\mathbf{x}_m$ and $\mathbf{y}_m$ respectively, and $w_{m,i}$ are the strictly positive weights of a perceptual weighting function $\mathbf{w}_m = [w_{m,0}, ..., w_{m,N-1}]^T \in \mathbb{R}^N$. The perceptual weighting function depends on the masking threshold $\mathbf{t}_m$ and is obeing the symmetry property $w_{m,i} = w_{m,N-i}$ for $i = 1, 2, ..., \frac{N}{2} - 1$ (see subsection 3.2.3).

Optimization problem (3.6) can then be rewritten as

$$\mathbf{v}_m^* = \underset{\mathbf{v}_m \in \mathbb{R}^N}{\arg\min} \ \frac{1}{2} (\mathbf{y}_m - \mathbf{x}_m)^T \underbrace{\mathbf{D}^H \mathbf{W}_m \mathbf{D}}_{\triangleq \mathbf{Q}_m} (\mathbf{y}_m - \mathbf{x}_m) \tag{3.7}$$



(a) (b)

(c) (d)

(e) (f)

(g)

Figure 3.2: Different steps in the computation of the global masking threshold using the ISO/IEC 11172-3 MPEG-1 Layer 1 psychoacoustic model 1 : (a)-(b) Time domain and normalized frequency domain representations of the input audio signal (c)-(d) Tonal maskers (circles), non-tonal maskers (squares) and input frequency spectrum (dotted line) (e)-(f) Individual masking thresholds related to tonal and non-tonal maskers respectively (g) Global masking threshold (solid line) and input frequency spectrum (dotted line). Figure adapted from [8].

where $\mathbf{D} \in \mathbb{C}^{N \times N}$ is the unitary Discrete Fourier Transform (DFT) matrix

$$\mathbf{D} = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & e^{-j\omega_1} & e^{-j\omega_2} & \dots & e^{-j\omega_{N-1}} \\ 1 & e^{-j\omega_2} & e^{-j\omega_4} & \dots & e^{-j\omega_{2(N-1)}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & e^{-j\omega_{N-1}} & e^{-j\omega_{2(N-1)}} & \dots & e^{-j\omega_{(N-1)(N-1)}} \end{bmatrix} \tag{3.8}$$

and $\mathbf{W}_m \in \mathbb{R}^{N \times N}$ is a diagonal weighting matrix with diagonal elements $w_{m,i}$. By inserting the Hammerstein model input-output relation (3.3), the optimization problem (3.7) can be rewritten as follows,

$$\begin{aligned} \mathbf{v}_m^* &= \arg\min_{\mathbf{v}_m \in \mathbb{R}^N} \; f_0(\mathbf{v}_m) \\ &= \arg\min_{\mathbf{v}_m \in \mathbb{R}^N} \; \frac{1}{2} \left(\mathbf{H}_m \mathbf{g}(\mathbf{v}_m) + \tilde{\mathbf{H}}_m \mathbf{g}(\mathbf{v}_{m-1}^*) - \mathbf{x}_m\right)^T \mathbf{Q}_m \\ &\qquad\qquad\qquad \left(\mathbf{H}_m \mathbf{g}(\mathbf{v}_m) + \tilde{\mathbf{H}}_m \mathbf{g}(\mathbf{v}_{m-1}^*) - \mathbf{x}_m\right) \\ &= \arg\min_{\mathbf{v}_m \in \mathbb{R}^N} \; \frac{1}{2} \, \mathbf{g}(\mathbf{v}_m)^T \underbrace{\mathbf{H}_m^T \mathbf{Q}_m \mathbf{H}_m}_{\triangleq \mathbf{A}_m} \mathbf{g}(\mathbf{v}_m) \\ &\qquad\qquad + \underbrace{\left(\mathbf{H}_m^T \mathbf{Q}_m (\tilde{\mathbf{H}}_m \mathbf{g}(\mathbf{v}_{m-1}^*) - \mathbf{x}_m)\right)}_{\triangleq \mathbf{b}_m}^T \mathbf{g}(\mathbf{v}_m). \end{aligned} \tag{3.9}$$

In general, the objective function $f_0(\mathbf{v}_m)$ of this optimization problem is not convex in the optimization variable $\mathbf{v}_m$. In Section 3.3, three distinct optimization methods for efficiently solving optimization problem (3.9) are proposed, each of these being specifically designed for a given class of memoryless nonlinearities $\mathbf{g}$ in the Hammerstein loudspeaker model.

3.2.3 Perceptual Weighting Function

The rationale behind applying perceptual weights in the summation (3.6) is the fact that distortion at different frequencies is differently perceived, and that the relative perceptibility is partly signal-dependent. Two phenomena of human auditory perception are responsible for this:

- A first phenomenon is the *absolute threshold of hearing*, which is defined as the required intensity (dB) of a pure tone such that an average listener will just hear the tone in a noiseless environment. The absolute threshold of hearing is a function of the tone frequency and has been measured experimentally [9].
- A second phenomenon is *simultaneous masking*, where the presence of strong spectral energy (the masker) masks the simultaneous presence of weaker spectral energy (the maskee), or in other words, renders it imperceptible.

Combining both phenomena, the *instantaneous global masking threshold* $\mathbf{t}_m$ of the input audio signal $\mathbf{x}_m$ gives the amount of signal energy (dB) at each frequency that can be masked by the signal. As such, the masking threshold $\mathbf{t}_m$ gives an indication of the relative perceptibility of signal components at different frequencies.

Consider the input frame $\mathbf{x}_m$ to act as the masker, and $\mathbf{y}_m - \mathbf{x}_m$ as the maskee. By selecting the weights $w_{m,i}$ to be exponentially decreasing with the value of the global masking threshold $\mathbf{t}_m$ of the signal $\mathbf{x}_m$ at frequency i, i.e.

$$w_{m,i} = \begin{cases} 10^{-\alpha t_{m,i}} & \text{if } 0 \leq i \leq \frac{N}{2} \\ 10^{-\alpha t_{m,N-i}} & \text{if } \frac{N}{2} < i \leq N-1 \end{cases} \tag{3.10}$$

the objective function reflects the amount of perceptible distortion introduced. Appropriate values of the compression parameter α have been determined to lie in the range $[0.01, 0.04]$.

The instantaneous global masking threshold $\mathbf{t}_m$ of a given audio signal is calculated using part of the ISO/IEC 11172-3 MPEG-1 Layer 1 psychoacoustic model 1. A complete description of the operation of this psychoacoustic model is beyond the scope of this paper (we refer the reader to [10] and [11]). We outline the relevant steps [8] in the computation of $\mathbf{t}_m$ and illustrate the result of each step on an example audio signal (see Figure 3.2):

1. *Spectral analysis and SPL normalization*: In this step a high-resolution spectral estimate of the audio signal is calculated, with spectral components expressed in terms of sound pressure level (SPL). After a normalization operation and a Hann windowing operation on the input signal frame, the PSD estimate is obtained through a 512-point DFT. Figure 3.2(a) shows the time-domain input signal, Figure 3.2(b) shows the resulting spectral estimate.
2. *Identification of tonal and non-tonal maskers*: It is known from psychoacoustic research that the tonality of a masking component has an influence on its masking properties [12]. For this reason it is important to discriminate between tonal maskers (defined as local maxima of the signal spectrum) and non-tonal maskers. The output of the DFT is used to determine the relevant tonal and non-tonal maskers in the spectrum of the audio signal. In a first phase, tonal maskers are identified at local maxima of the PSD: energy from three adjacent spectral components centered at the local maximum is combined to form a single tonal masker. In a second phase, a single non-tonal masker per critical band is formed by addition of all the energy from the spectral components within the critical band that have not contributed to a tonal masker.
3. *Decimation of maskers*: In this step, the number of maskers is reduced using two criteria. First, any tonal or non-tonal masker below the absolute threshold of hearing is discarded. Next, any pair of maskers occurring within a distance of 0.5 Bark is replaced by the stronger of the

two. Figures 3.2(c) and 3.2(d) depict the identified tonal and non-tonal maskers respectively, after decimation.

4. *Calculation of individual masking thresholds*: an individual masking threshold is calculated for each masker in the decimated set of tonal and non-tonal maskers, using fixed psychoacoustic rules. Essentially, the individual masking threshold depends on the frequency, loudness level and tonality of the masker. Figure 3.2(e) and 3.2(f) show the individual masking thresholds associated with tonal and non-tonal maskers, respectively.
5. *Calculation of global masking threshold*: Finally, the global masking threshold is calculated by a power-additive combination of the tonal and non-tonal individual masking thresholds, and the absolute threshold of hearing. This is illustrated in Figure 3.2(g).

3.3 Optimization Methods

The efficient solution of the per-frame optimization problem (3.9) is crucial for the proposed loudspeaker precompensation algorithm to be applicable in real-time environments. In this section, three optimization methods for efficiently solving the considered optimization problem are proposed, each optimization method being specifically designed for a given class of memoryless nonlinearities $\mathbf{g}$ in the Hammerstein loudspeaker model.

3.3.1 Classes of Memoryless Nonlinearities

In order to define the considered classes of memoryless nonlinearities, the notions of *invertibility* and *smoothness* for memoryless nonlinearities are first introduced.

Definition 3.1 [Invertible memoryless nonlinearity] A memoryless nonlinearity $y = g(x)$ is *invertible* on the domain $D \subseteq \mathbb{R}$ if there exists a unique function $x = g^{-1}(y)$ for which

$$g^{-1}(g(x)) = x, \quad \forall\, x \in D. \tag{3.11}$$

The function $g^{-1}(y)$ is called the *inverse memoryless nonlinearity* of $g(x)$ on the domain D. The output $\mathbf{g^{-1}}(\mathbf{y}_m)$ of the inverse memoryless nonlinearity for a given frame $\mathbf{y}_m$ is constructed as

$$\mathbf{g^{-1}}(\mathbf{y}_m) = [g^{-1}(y_{m,1}), ..., g^{-1}(y_{m,N})]^T. \tag{3.12}$$

□

Definition 3.2 [Smooth memoryless nonlinearity] A memoryless nonlinearity $y = g(x)$ is *smooth* on the domain $D \subseteq \mathbb{R}$ if its derivatives of all orders exist and are continuous on D. □

Based on the above definitions, the following three classes of memoryless nonlinearities[1] can be determined:

- Class I is formed by the *invertible memoryless nonlinearities*, such as the one depicted in Figure 3.3(a).
- Class II is formed by the *non-invertible smooth memoryless nonlinearities*, such as the one depicted in Figure 3.3(b).
- Class III is formed by the *non-invertible non-smooth hard clipping memoryless nonlinearities*, such as the one depicted in Figure 3.3(c). This class of memoryless nonlinearities has a characteristic of the form

$$g(x) = \begin{cases} x, & |x| \leq U \\ \mathrm{sgn}(x)U, & |x| > U \end{cases} \tag{3.13}$$

with U defined as the *clipping level*.

For each of these classes of memoryless nonlinearities, a specific optimization method is proposed for solving the optimization problem (3.9) in a fast and reliable way.

3.3.2 Class I: Invertible Memoryless Nonlinearities

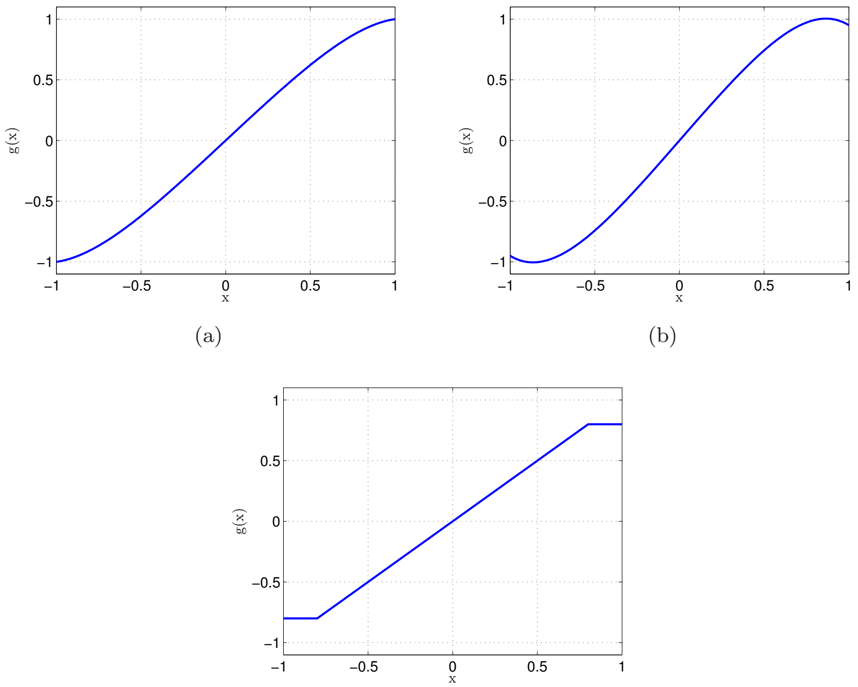
In case the memoryless nonlinearity $\mathbf{g}$ in the Hammerstein model is invertible, the precompensation of the linear FIR filter $\mathbf{h}$ and the precompensation of the memoryless nonlinearity $\mathbf{g}$ can be decoupled. In a first step, the linear FIR filter can be precompensated for, resulting in the nonlinearly transformed frame $\mathbf{g}(\mathbf{v}_m^*)$. In a second step, the memoryless nonlinearity $\mathbf{g}$ can be precompensated for by applying the inverse memoryless nonlinearity $\mathbf{g}^{-\mathbf{1}}$ to $\mathbf{g}(\mathbf{v}_m^*)$ in order to compute the precompensated input frame $\mathbf{v}_m^*$, i.e.

$$\mathbf{v}_m^* = \mathbf{g}^{-\mathbf{1}}(\mathbf{g}(\mathbf{v}_m^*)). \tag{3.14}$$

In order to carry out the first precompensation step, the objective function in (3.9) can be seen to be a strictly convex quadratic function of the nonlinearly transformed variable $\mathbf{g}(\mathbf{v}_m)$, i.e.

$$\mathbf{g}(\mathbf{v}_m^*) = \underset{\mathbf{g}(\mathbf{v}_m) \in \mathbb{R}^N}{\arg\min} \; f_1(\mathbf{g}(\mathbf{v}_m))$$

[1] In the remainder of this paper, all considered memoryless nonlinearities are assumed to operate on the domain $D = [-1, 1]$.



(a)

(b)

(c)

Figure 3.3: Examples for different classes of memoryless nonlinearities on the domain $D = [-1, 1]$: (a) Invertible memoryless nonlinearity $g(x) = 1.3x - 0.2x^3 - 0.1x^5$ (b) Non-invertible smooth memoryless nonlinearity $g(x) = 1.6x - 0.4x^3 - 0.25x^5$ (c) Non-invertible non-smooth hard clipping memoryless nonlinearity $g(x) = \max(\min(x, 0.8), -0.8)$.

$$= \underset{\mathbf{g}(\mathbf{v}_m)\in\mathbb{R}^N}{\arg\min}\ \frac{1}{2}\,\mathbf{g}(\mathbf{v}_m)^T\,\mathbf{A}_m\,\mathbf{g}(\mathbf{v}_m) + \mathbf{b}_m^T\,\mathbf{g}(\mathbf{v}_m) \tag{3.15}$$

with $\mathbf{A}_m \succ 0$. Indeed, as it was shown that $\mathbf{Q}_m \succ 0$ in [8], it follows that $\mathbf{A}_m = \mathbf{H}_m^T\mathbf{Q}_m\mathbf{H}_m \succ 0$, since congruence transformations have the property of preserving positive definiteness.

The necessary and sufficient condition for $\mathbf{g}(\mathbf{v}_m^*)$ to be the unique global minimizer of this strictly convex optimization problem, is

$$\nabla f_1(\mathbf{g}(\mathbf{v}_m^*)) = \mathbf{0} \tag{3.16}$$

from which the following closed form solution $\mathbf{g}(\mathbf{v}_m^*)$ can be derived,

$$\begin{aligned}\mathbf{g}(\mathbf{v}_m^*) &= -\mathbf{A}_m^{-1}\mathbf{b}_m\\ &= -(\mathbf{H}_m^T\mathbf{Q}_m\mathbf{H}_m)^{-1}(\mathbf{H}_m^T\mathbf{Q}_m(\tilde{\mathbf{H}}_m\mathbf{g}(\mathbf{v}_{m-1}^*) - \mathbf{x}_m))\\ &= -\mathbf{H}_m^{-1}(\tilde{\mathbf{H}}_m\mathbf{g}(\mathbf{v}_{m-1}^*) - \mathbf{x}_m)\end{aligned} \tag{3.17}$$

where the symmetry of $\mathbf{Q}_m$ is used and the invertibility of $\mathbf{H}_m$ is assumed to establish the last equality. It can be easily shown that $f_1(\mathbf{g}(\mathbf{v}_m^*)) = 0$, implying that a perfect precompensation $\mathbf{y}_m^* = \mathbf{x}_m$ is achieved. Note that the perceptual weighting $\mathbf{W}_m$ of the frequency domain error terms (which are all zero in the solution) is unable to affect this solution, explaining the absence of $\mathbf{Q}_m$ in (3.17).

3.3.3 Class II: Non-Invertible Smooth Memoryless Nonlinearities

In case the memoryless nonlinearity $\mathbf{g}$ in the Hammerstein model is non-invertible, the precompensation of the linear FIR filter $\mathbf{h}$ and the precompensation of the memoryless nonlinearity $\mathbf{g}$ should be performed simultaneously by solving optimization problem (3.9). From the smoothness property of $\mathbf{g}$ and its constituting basis functions $\boldsymbol{\psi}$, it follows that for any $\mathbf{x}_m \in D^N$ the gradient $\nabla\mathbf{g}(\mathbf{x}_m) \in \mathbb{R}^N$ can be computed as

$$\begin{aligned}\nabla\mathbf{g}(\mathbf{x}_m) &= [\nabla g(x_{m,1}), ..., \nabla g(x_{m,N})]^T\\ &= \nabla\boldsymbol{\Psi}(\mathbf{x}_m)\mathbf{c}\end{aligned} \tag{3.18}$$

where the matrix $\nabla\boldsymbol{\Psi}(\mathbf{x}_m) \in \mathbb{R}^{N\times P}$ is defined as

$$\nabla\boldsymbol{\Psi}(\mathbf{x}_m) = \begin{bmatrix}\nabla\psi_1(x_{m,1}) & \dots & \nabla\psi_P(x_{m,1})\\ \vdots & \ddots & \vdots\\ \nabla\psi_1(x_{m,N}) & \dots & \nabla\psi_P(x_{m,N})\end{bmatrix}. \tag{3.19}$$

The proposed optimization method for solving the non-convex optimization problem (3.9) is an iterative *gradient method.* Introducing the notation $\mathbf{v}_m^k$ for the kth iterate of the mth frame, the $(k+1)$th iteration of the iterative gradient method consists in taking a step along the negative gradient direction,

$$\mathbf{v}_m^{k+1} = \mathbf{v}_m^k - s_m^k \nabla f_0(\mathbf{v}_m^k) \tag{3.20}$$

where s_m^k is the stepsize and the gradient $\nabla f_0(\mathbf{v}_m^k)$ is computed as

$$\begin{aligned}\nabla f_0(\mathbf{v}_m^k) = \mathrm{diag}(\nabla \mathbf{g}(\mathbf{v}_m^k)) \Big[\mathbf{H}_m^T \mathbf{Q}_m \Big(&\mathbf{H}_m \mathbf{g}(\mathbf{v}_m^k) \\ &+ \tilde{\mathbf{H}}_m \mathbf{g}(\mathbf{v}_{m-1}^*) - \mathbf{x}_m\Big)\Big]\end{aligned} \tag{3.21}$$

and where $\nabla \mathbf{g}(\mathbf{v}_m^k)$ is established using (3.18)-(3.19).

The stepsize s_m^k is determined using a backtracking line search for satisfying the Armijo sufficient decrease condition [13],

$$f_0(\mathbf{v}_m^k - s_m^k \nabla f_0(\mathbf{v}_m^k)) \leq f_0(\mathbf{v}_m^k) - \beta s_m^k \|\nabla f_0(\mathbf{v}_m^k)\|_2^2 \tag{3.22}$$

with $\beta \in (0, \frac{1}{2})$ the relaxation of the gradient. Starting from $s_m^k = 1$, successive reductions of s_m^k by a factor $\eta \in (0, 1)$ are performed until the condition (3.22) holds. The resulting algorithm is given in Algorithm 4.

3.3.4 Class III: Non-Invertible Hard Clipping Memoryless Nonlinearities

In case the memoryless nonlinearity $\mathbf{g}$ in the Hammerstein model is a non-invertible hard clipping memoryless nonlinearity with clipping level U as defined in (3.13), optimization problem (3.9) can be rewritten as follows,

$$\begin{aligned}\mathbf{v}_m^* &= \underset{\mathbf{v}_m \in \mathbb{R}^N}{\arg\min}\ f_1(\mathbf{v}_m) \quad \text{s.t.} \quad -\mathbf{u} \leq \mathbf{v}_m \leq \mathbf{u} \\ &= \underset{\mathbf{v}_m \in \mathbb{R}^N}{\arg\min}\ \frac{1}{2}\, \mathbf{v}_m^T\, \mathbf{A}_m\, \mathbf{v}_m + \mathbf{b}_m^T\, \mathbf{v}_m \\ &\qquad\qquad\qquad\qquad \text{s.t.} \quad -\mathbf{u} \leq \mathbf{v}_m \leq \mathbf{u}\end{aligned} \tag{3.23}$$

where $\mathbf{u} = U\mathbf{1}_N$ contains the per-sample clipping levels and $\mathbf{1}_N \in \mathbb{R}^N$ is an all ones vector. Note that by constraining $\mathbf{v}_m$ to be within the linear range $[-U, U]$ of the memoryless nonlinearity $\mathbf{g}$, it is possible to introduce the equivalence $\mathbf{g}(\mathbf{v}_m) = \mathbf{v}_m$ in the objective function. The strictly convex quadratic objective function ($\mathbf{A}_m \succ 0$) and the affine constraints make optimization problem (3.23) a strictly convex *quadratic program.*

The proposed optimization method for solving the strictly convex optimization problem (3.23) is an iterative *projected gradient method.* In each iteration of a projected gradient method, first a step is taken along the negative gradient direction of the objective function, after which the result is orthogonally

Algorithm 4 Precompensation of a Hammerstein loudspeaker model with class II memoryless nonlinearity using the gradient optimization method

```
Input $\mathbf{x}_m \in \mathbb{R}^N$, $\mathbf{v}^*_{m-1} \in \mathbb{R}^N$, $\mathbf{h} \in \mathbb{R}^{L+1}$, $\mathbf{c} \in \mathbb{R}^P$, $\boldsymbol{\psi}(x) \in C(\mathbb{R} \to \mathbb{R}^P)$,
    $\mathbf{D} \in \mathbb{C}^{N\times N}$, $\alpha \in \mathbb{R}^+$, $\beta \in (0, \frac{1}{2})$, $\eta \in (0, 1)$, $K \in \mathbb{N}^+$
Output $\mathbf{v}^*_m \in \mathbb{R}^N$
Compute masking threshold $\mathbf{t}_m$ for $\mathbf{x}_m$ [using [11]]
Compute weights $\mathbf{w}_m$ [using (3.10)]
Construct $\mathbf{Q}_m = \mathbf{D}^H \text{diag}(\mathbf{w}_m)\mathbf{D}$
Construct $\mathbf{H}_m$ and $\tilde{\mathbf{H}}_m$ from $\mathbf{h}$ [using (3.4)-(3.5)]
Initialize $\mathbf{v}^0_m = \mathbf{x}_m$
Initialize $k = 0$
while $k < K$ do
    Compute $\nabla f_0(\mathbf{v}^k_m)$ [using (3.18)-(3.19) and (3.21)]
    Initialize $s^k_m = 1$
    while $f_0(\mathbf{v}^k_m - s^k_m \nabla f_0(\mathbf{v}^k_m)) > f_0(\mathbf{v}^k_m) - \beta s^k_m \|\nabla f_0(\mathbf{v}^k_m)\|^2_2$ do
        $s^k_m = \eta s^k_m$
    end while
    $\mathbf{v}^{k+1}_m = \mathbf{v}^k_m - s^k_m \nabla f_0(\mathbf{v}^k_m)$
    $k = k + 1$
end while
$\mathbf{v}^*_m = \mathbf{v}^k_m$
```

projected onto the convex feasible set, thereby maintaining feasibility of the iterates. A low computational complexity per iteration is the main asset of projected gradient methods, provided that the orthogonal projection onto the convex feasible set and the gradient of the objective function can easily be computed. The main steps in the $(k+1)$th iteration of the standard projected gradient method can be written as follows:

- Take a step of stepsize s^k_m along the negative gradient direction :

  $$\tilde{\mathbf{v}}^{k+1}_m = \mathbf{v}^k_m - s^k_m \nabla f_1(\mathbf{v}^k_m) \tag{3.24}$$

  where the gradient $\nabla f_1(\mathbf{v}^k_m)$ is computed as

  $$\nabla f_1(\mathbf{v}^k_m) = \mathbf{H}^T_m \mathbf{Q}_m(\mathbf{H}_m \mathbf{v}^k_m + \tilde{\mathbf{H}}_m \mathbf{v}^*_{m-1} - \mathbf{x}_m). \tag{3.25}$$

- Project $\tilde{\mathbf{v}}^{k+1}_m$ orthogonally onto the convex feasible set $\Omega = \{\mathbf{v}_m \in \mathbb{R}^N | -\mathbf{u} \leq \mathbf{v}_m \leq \mathbf{u}\}$ of optimization problem (3.23),

  $$\mathbf{v}^{k+1}_m = \Pi_\Omega(\tilde{\mathbf{v}}^{k+1}_m) = \underset{\mathbf{v}_p \in \Omega}{\arg\min} \frac{1}{2}\|\mathbf{v}_p - \tilde{\mathbf{v}}^{k+1}_m\|^2_2 \tag{3.26}$$

  An orthogonal projection $\Pi_\Omega(\tilde{\mathbf{v}}^{k+1}_m)$ onto Ω can be shown to correspond to performing a componentwise hard clipping operation with clipping level U,

Algorithm 5 Precompensation of a Hammerstein loudspeaker model with class III memoryless nonlinearity using the optimal projected gradient optimization method

```
Input $\mathbf{x}_m \in \mathbb{R}^N$, $\mathbf{v}^*_{m-1} \in \mathbb{R}^N$, $\mathbf{h} \in \mathbb{R}^{L+1}$, $\mathbf{D} \in \mathbb{C}^{N\times N}$, $U \in (0,1]$, $\alpha \in \mathbb{R}^+$,
    $\gamma^0_m \in (0,1)$, $K \in \mathbb{N}^+$
Output $\mathbf{v}^*_m \in \mathbb{R}^N$
Compute masking threshold $\mathbf{t}_m$ for $\mathbf{x}_m$ [using [11]]
Compute weights $\mathbf{w}_m$ [using (3.10)]
Construct convex feasible set $\Omega = \{\mathbf{v}_m \in \mathbb{R}^N | -\mathbf{u} \leq \mathbf{v}_m \leq \mathbf{u}\}$
Construct $\mathbf{Q}_m = \mathbf{D}^H \mathrm{diag}(\mathbf{w}_m)\mathbf{D}$
Construct $\mathbf{H}_m$ and $\tilde{\mathbf{H}}_m$ from $\mathbf{h}$ [using (3.4)-(3.5)]
Establish Lipschitz constant $C_m$ [using (3.35)]
Establish convexity parameter $\mu_m$ [using (3.36)]
Compute $\kappa_m = \frac{C_m}{\mu_m}$
Initialize $\mathbf{v}^0_m = \mathbf{c}^0_m = \mathbf{x}_m$
Initialize $k = 0$
while $k < K$ do
    $\tilde{\mathbf{v}}^{k+1}_m = \mathbf{c}^k_m - \frac{1}{C_m}\nabla f_1(\mathbf{c}^k_m)$ [using (3.25)]
    $\mathbf{v}^{k+1}_m = \Pi_\Omega(\tilde{\mathbf{v}}^{k+1}_m)$    [using (3.27)]
    Calculate $\gamma^{k+1}_m$ from $(\gamma^{k+1}_m)^2 = (1-\gamma^{k+1}_m)(\gamma^k_m)^2 + \kappa_m \gamma^{k+1}_m$
    $\delta^k_m = \frac{\gamma^k_m(1-\gamma^k_m)}{(\gamma^k_m)^2+\gamma^{k+1}_m}$
    $\mathbf{c}^{k+1}_m = \mathbf{v}^{k+1}_m + \delta^k_m(\mathbf{v}^{k+1}_m - \mathbf{v}^k_m)$
    $k = k + 1$
end while
$\mathbf{v}^*_m = \mathbf{v}^k_m$
```

i.e. $\Pi_\Omega(\tilde{\mathbf{v}}^{k+1}_m) = \mathbf{g}(\tilde{\mathbf{v}}^{k+1}_m) = [g(\tilde{v}^{k+1}_{m,1}), ..., g(\tilde{v}^{k+1}_{m,N})]^T$, with

$$g(\tilde{v}^{k+1}_{m,i}) = \begin{cases} \tilde{v}^{k+1}_{m,i}, & |\tilde{v}^{k+1}_{m,i}| \leq U \\ \mathrm{sgn}(\tilde{v}^{k+1}_{m,i})U, & |\tilde{v}^{k+1}_{m,i}| > U. \end{cases} \tag{3.27}$$

Variants of the standard projected gradient method have been proposed which have better convergence properties. Here, a projected gradient optimization method is adopted that reaches an optimal convergence for the class of convex optimization problems with *strongly convex* objective functions. This method was first proposed in [14] and has been applied in diverse applications, e.g. real-time clipping of audio signals [15]. Let us assume knowledge of the Lipschitz constant C_m of the gradient ∇f_1 and the convexity parameter μ_m of f_1 on the set Ω. Then, in each iteration k of the optimal projected gradient method, a standard projected gradient step is performed on a weighted sum $\mathbf{c}^k_m$ of two previous feasible iterates $\mathbf{v}^k_m$ and $\mathbf{v}^{k-1}_m$,

$$\mathbf{v}^{k+1}_m = \Pi_\Omega\left(\mathbf{c}^k_m - \frac{1}{C_m}\nabla f_1(\mathbf{c}^k_m)\right) \tag{3.28}$$

where

$$\mathbf{c}_m^k = \mathbf{v}_m^k + \delta_m^k(\mathbf{v}_m^k - \mathbf{v}_m^{k-1}) \tag{3.29}$$

where δ_m^k is computed as

$$\delta_m^k = \frac{\gamma_m^{k-1}(1-\gamma_m^{k-1})}{(\gamma_m^{k-1})^2 + \gamma_m^k} \tag{3.30}$$

and where γ_m^k is computed as the solution of

$$(\gamma_m^k)^2 = (1-\gamma_m^k)(\gamma_m^{k-1})^2 + \kappa_m \gamma_m^k \tag{3.31}$$

and $\kappa_m = \frac{C_m}{\mu_m}$ is the condition number.

In order to establish C_m and μ_m for optimization problem (3.23), the next two lemmas are proposed.

Lemma 3 *([14]) Let function f_1 be twice continuously differentiable on set Ω. The gradient ∇f_1 is Lipschitz continuous on set Ω with Lipschitz constant C if and only if*

$$\nabla^2 f_1(\mathbf{z}) \preceq C\mathbf{I}\ ,\ \forall \mathbf{z} \in \Omega \tag{3.32}$$

Lemma 4 *([14]) Let function f_1 be twice continuously differentiable on set Ω. The function f_1 is strongly convex on set Ω with convexity parameter μ if and only if there exists $\mu > 0$ such that*

$$\nabla^2 f_1(\mathbf{z}) \succeq \mu\mathbf{I}\ ,\ \forall \mathbf{z} \in \Omega \tag{3.33}$$

Let us denote and order the eigenvalues of $\mathbf{A}_m$ as follows,

$$\lambda_1(\mathbf{A}_m) \leq \lambda_2(\mathbf{A}_m) \leq \cdots \leq \lambda_{N-1}(\mathbf{A}_m) \leq \lambda_N(\mathbf{A}_m). \tag{3.34}$$

Using Lemma 3.1 and Lemma 3.2, it is then shown that the Lipschitz constant C_m can be computed as the largest eigenvalue of $\mathbf{A}_m$,

$$C_m = \lambda_N(\mathbf{A}_m) = \lambda_N(\mathbf{H}_m^T\mathbf{Q}_m\mathbf{H}_m) \tag{3.35}$$

and that the convexity parameter μ_m can be computed as the smallest eigenvalue of $\mathbf{A}_m$,

$$\mu_m = \lambda_1(\mathbf{A}_m) = \lambda_1(\mathbf{H}_m^T\mathbf{Q}_m\mathbf{H}_m). \tag{3.36}$$

The resulting algorithm is given as Algorithm 5.

3.3.5 Algorithmic Complexity Bounds

It is possible to derive polynomial upper and lower bounds on the *algorithmic complexity*, i.e. the number of necessary iterations of the optimization method proposed in subsection 3.3.4 as a function of the solution accuracy. For the class of convex optimization problems with strongly convex objective functions, bounds on the algorithmic complexity of the projected gradient method used in Algorithm 5 are derived as follows:

Theorem 3.3 ([14]) *Let f_1 be a strongly convex, continuously differentiable function with Lipschitz constant C_m and convexity parameter $\mu_m = \frac{C_m}{\kappa_m}$ (where the condition number $\kappa_m > 1$), and let Ω be a convex feasible set. Then the minimum and maximum number of necessary iterations of the optimal projected gradient method described in Algorithm 5 to find a solution $\mathbf{v}_m^k$ satisfying $f_1(\mathbf{v}_m^k) - f_1(\mathbf{v}_m^*) \leq \epsilon$ are given by*

$$k_m^{\min} = \frac{\sqrt{\kappa_m} - 1}{4} \log\left(\frac{\mu_m ||\mathbf{v}_m^0 - \mathbf{v}_m^*||^2}{2\epsilon}\right) \tag{3.37}$$

$$k_m^{\max} = \sqrt{\kappa_m} \log\left(\frac{C_m ||\mathbf{v}_m^0 - \mathbf{v}_m^*||^2}{\epsilon}\right). \tag{3.38}$$

It has been shown that congruence transformations like $\mathbf{Q}_m \to \mathbf{H}_m^T \mathbf{Q}_m \mathbf{H}_m$ do preserve the signs but not the magnitudes of the matrix eigenvalues. Interestingly, bounds on the change in magnitude of the eigenvalues under a congruence transformation have been established,

Theorem 3.4 *([16]) Let $\mathbf{Q}_m \in \mathbb{R}^{N\times N}$ be symmetric and $\mathbf{H}_m \in \mathbb{R}^{N\times N}$ nonsingular. Let the eigenvalues of $\mathbf{Q}_m$ and $\mathbf{H}_m^T \mathbf{Q}_m \mathbf{H}_m$ be arranged in nondecreasing order* (3.34)*. Let $\sigma_1(\mathbf{H}_m) \geq \cdots \geq \sigma_N(\mathbf{H}_m) > 0$ be the singular values of $\mathbf{H}_m$. Then each eigenvalue $\lambda_i(\mathbf{H}_m^T \mathbf{Q}_m \mathbf{H}_m)$, $i = 1, ..., N$, is bounded as follows,*

$$\sigma_N^2(\mathbf{H}_m)\lambda_i(\mathbf{Q}_m) \leq \lambda_i(\mathbf{H}_m^T \mathbf{Q}_m \mathbf{H}_m) \leq \sigma_1^2(\mathbf{H}_m)\lambda_i(\mathbf{Q}_m). \tag{3.39}$$

Corollary 3.5 *The condition number $\kappa(\mathbf{H}_m^T \mathbf{Q}_m \mathbf{H}_m) = \frac{\lambda_N(\mathbf{H}_m^T \mathbf{Q}_m \mathbf{H}_m)}{\lambda_1(\mathbf{H}_m^T \mathbf{Q}_m \mathbf{H}_m)}$ is bounded as follows,*

$$\kappa(\mathbf{H}_m^T \mathbf{Q}_m \mathbf{H}_m) \geq \max\left(1, \kappa^{-1}(\mathbf{H}_m^T \mathbf{H}_m)\kappa(\mathbf{Q}_m)\right) \tag{3.40}$$

$$\kappa(\mathbf{H}_m^T \mathbf{Q}_m \mathbf{H}_m) \leq \kappa(\mathbf{H}_m^T \mathbf{H}_m)\kappa(\mathbf{Q}_m). \tag{3.41}$$

The previous theorem and its corollary now allow to rewrite the lower and upper iteration bounds (3.37)-(3.38) as a function of the extreme eigenvalues and singular values of $\mathbf{H}_m$ and $\mathbf{Q}_m$.

Theorem 3.6 *The minimum and maximum number of necessary iterations of the optimal projected gradient method described in Algorithm 5 to find* $\mathbf{v}_m^k$ *satisfying* $f_1(\mathbf{v}_m^k) - f_1(\mathbf{v}_m^*) \leq \epsilon$ *are given by*

$$k_m^{\min} = \frac{\sqrt{\max\left(1, \kappa^{-1}(\mathbf{H}_m^T \mathbf{H}_m)\kappa(\mathbf{Q}_m)\right)} - 1}{4} \log\left(\frac{\lambda_1(\mathbf{H}_m^T \mathbf{H}_m)\lambda_1(\mathbf{Q}_m)\,||\mathbf{v}_m^0 - \mathbf{v}_m^*||^2}{2\epsilon}\right) \tag{3.42}$$

$$k_m^{\max} = \sqrt{\kappa(\mathbf{H}_m^T \mathbf{H}_m)\kappa(\mathbf{Q}_m)} \log\left(\frac{\lambda_N(\mathbf{H}_m^T \mathbf{H}_m)\lambda_N(\mathbf{Q}_m)\,||\mathbf{v}_m^0 - \mathbf{v}_m^*||^2}{\epsilon}\right). \tag{3.43}$$

Proof : In the lower iteration bound (3.37) established in Theorem 3.3, incorporate the lower bound (3.39) on $\mu_m = \lambda_1(\mathbf{H}_m^T \mathbf{Q}_m \mathbf{H}_m)$ established in Theorem 3.4, and the lower bound (3.40) on $\kappa_m = \kappa(\mathbf{H}_m^T \mathbf{Q}_m \mathbf{H}_m)$ established in Corollary 3.5. This yields the lower iteration bound (3.42). Analogously, in the upper iteration bound (3.38) established in Theorem 3.3, incorporate the upper bound (3.39) on $C_m = \lambda_N(\mathbf{H}_m^T \mathbf{Q}_m \mathbf{H}_m)$ established in Theorem 3.4, and the upper bound (3.41) on $\kappa_m = \kappa(\mathbf{H}_m^T \mathbf{Q}_m \mathbf{H}_m)$ established in Corollary 3.5. This yields the upper iteration bound (3.43). □

3.4 Audio Quality Evaluation

The proposed loudspeaker precompensation algorithm is evaluated in terms of the resulting audio quality. Subsection 3.4.1 reports audio quality evaluation experiments using synthetic Hammerstein loudspeaker models, comprising different classes of memoryless nonlinearities. Subsection 3.4.2 reports audio quality evaluation experiments using Hammerstein loudspeaker models identified on real loudspeakers. The identification procedure of the Hammerstein loudspeaker models is discussed in Subsection 3.4.3.

3.4.1 Audio Quality Evaluation Using Synthetic Hammerstein Loudspeaker Models

Evaluation experiments were performed using two synthetic Hammerstein loudspeaker models. The first Hammerstein model consists of a non-invertible smooth memoryless nonlinearity (class II), followed by a linear FIR filter. The second Hammerstein model consists of a non-invertible hard clipping memoryless nonlinearity (class III), followed by the same linear FIR filter. The following specifications apply:

Table 3.1: Audio excerpts database used for comparative audio quality evaluation (16 bit mono at 44.1 kHz).

Nr.	Name	Texture	Composition	Style	Duration [s]	Sample$_{\text{start}}$	Sample$_{\text{end}}$	Origin
1	poulenc.wav	polyphonic	instrumental	classical	17.8	400000	1183000	[17]
2	rhcp.wav	polyphonic	instrumental	rock	9.8	468996	900000	[18]
3	pierle.wav	polyphonic	instrumental+vocal	pop	11.7	2234000	2750000	[19]
4	chopin.wav	monophonic	instrumental	classical	17.8	50000	836200	[20]
5	kraftw.wav	polyphonic	instrumental	electronic	17.2	7480000	8240000	[21]
6	crefsax.wav	monophonic	instrumental	classical	10.9	1	479026	[22]
7	grefcla.wav	monophonic	instrumental	classical	6.9	1	302534	[22]
8	arefcla.wav	monophonic	instrumental	classical	7.4	1	302534	[22]

- Class II memoryless nonlinearity: $P = 3$ basis functions $\boldsymbol{\psi}(x) = [x\ x^3\ x^5]^T$, a corresponding coefficient vector $\mathbf{c} = [1.6\ \ -0.4\ \ -0.25]^T$, depicted in Figure 3.3(b).
- Class III memoryless nonlinearity: of the form (3.13), with clipping level $U = 0.8$, depicted in Figure 3.3(c).
- Linear FIR filter: $L = 128$, impulse response $h[n]$, designed using the frequency sampling method `fir2` in Matlab, having a required magnitude response $[1\ 0.95\ 0.75\ 0.50\ 0.20\ 0]^T$ for the frequencies $[0\ 0.2\ 0.4\ 0.6\ 0.8\ 1]^T \times f_{\text{Nyquist}}$.

A test database consisting of 8 audio excerpts was compiled (see Table 3.1 for details). Each audio signal in the test database was fed into each of the considered Hammerstein loudspeaker models, once with and once without performing precompensation. The following settings were used in the precompensation:

- First Hammerstein model: precompensation using Algorithm 4, with $N = 512$, $\alpha = 0.01$, $\beta = 0.1$, $\eta = 0.6$, and $K = 500$.
- Second Hammerstein model: precompensation using Algorithm 5, with $N = 512$, $\alpha = 0.01$, $\gamma_m^0 = \sqrt{\frac{\mu_m}{C_m}}$, and $K = 500$.

The objective audio quality improvement for each audio signal was assessed by computing the ΔODG measure,

$$\Delta\text{ODG} = \text{ODG}(\mathbf{x}, \mathbf{y}^*) - \text{ODG}(\mathbf{x}, \mathbf{y}) \tag{3.44}$$

where $\mathbf{x}$ is the input signal, $\mathbf{y}$ is the output signal when no precompensation is applied, $\mathbf{y}^*$ is the output signal when precompensation is applied, and $\text{ODG}(\mathbf{r}, \mathbf{d})$ is an objective measure [22] that predicts the audio quality of a signal $\mathbf{d}$ with respect to a signal $\mathbf{r}$ on a scale of $[0, -4]$, where 0 corresponds to an imperceptible degradation, and -4 corresponds to a very annoying degradation.

The simulations were performed for four distinct average amplitude levels of the input signals, selected such that the corresponding $\text{ODG}(\mathbf{x}, \mathbf{y}) = \{-0.5, -1.0, -1.5, -2.0\}$. This signal-specific selection of the average amplitude levels ensures a uniform audio quality degradation for different output signals when no precompensation is applied.

The resulting ΔODG scores for the first and second Hammerstein models are shown in Figure 3.4 and Figure 3.5, respectively. For the first Hammerstein model in Figure 3.4, a positive audio quality improvement is observed for all audio excerpts, and this for all but one considered average amplitude level. For most audio excerpts, increasing audio quality improvement scores are observed for increasing average amplitude levels. Indeed, it is exactly at higher amplitude levels that the Hammerstein memoryless nonlinearity is severely affecting the audio signal, and that precompensation is capable of improving the resulting audio quality considerably.

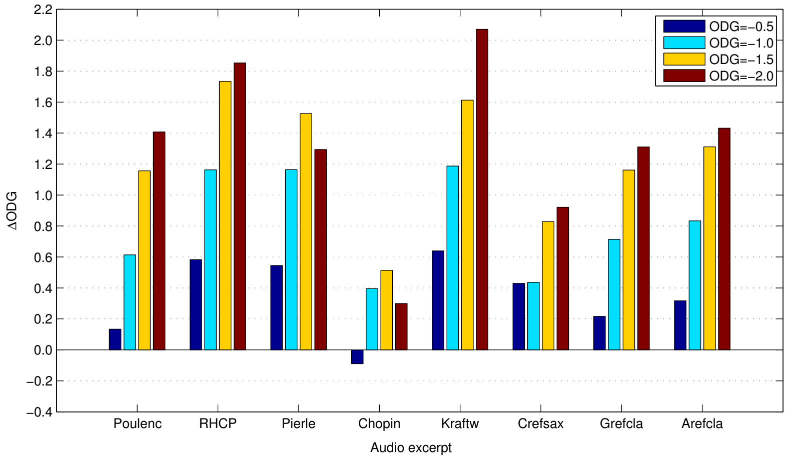


Figure 3.4: Precompensation of a Hammerstein model with non-invertible smooth memoryless nonlinearity: objective audio quality improvement scores for different audio excerpts, at four distinct average amplitude levels corresponding to uncompensated input ODGs= $\{-0.5, -1.0, -1.5, -2.0\}$.

For the second Hammerstein model in Figure 3.5, a similar pattern is observed with predominantly positive audio quality improvement scores that are increasing with the average amplitude levels. On average, the resulting audio quality improvement is seen to be slightly lower than for the first Hammerstein model. This difference can be attributed to the non-smoothness of the memoryless nonlinearity, which makes the second Hammerstein model more difficult to precompensate for, compared to the first Hammerstein model which comprises a smooth memoryless nonlinearity.

3.4.2 Audio Quality Evaluation Using Identified Hammerstein Loudspeaker Models

Evaluation experiments were performed using two different loudspeakers having the following specifications:

- Loudspeaker 1: Boss MA-12 Micro Monitor, full-range, impedance 6Ω.
- Loudspeaker 2: Dell Latitude E6400 laptop built-in speaker.

The loudspeakers were modeled using Hammerstein models consisting of a memoryless nonlinearity with $P = 3$ basis functions $\boldsymbol{\psi}(x) = [x\ x^3\ x^5]^T$ and a linear FIR filter of order $L = 128$. The identification of the loudspeaker

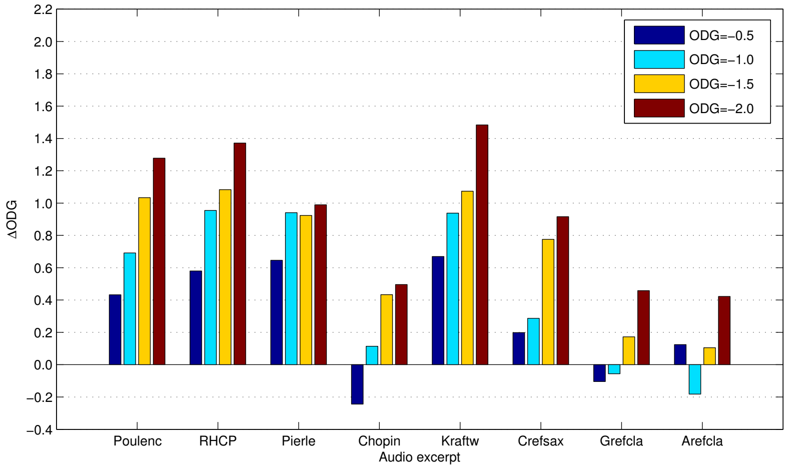


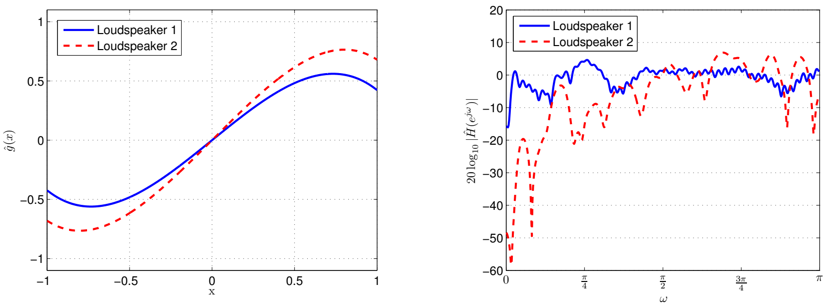
Figure 3.5: Precompensation of a Hammerstein model with non-invertible hard clipping memoryless nonlinearity: objective audio quality improvement scores for different audio excerpts, at four distinct average amplitude levels corresponding to uncompensated input ODGs= $\{-0.5, -1.0, -1.5, -2.0\}$.

Hammerstein model parameters $\mathbf{h} \in \mathbb{R}^{L+1}$ and $\mathbf{c} \in \mathbb{R}^{P}$ was performed using the normalized iterative algorithm proposed in [23], and detailed in Subsection 3.4.3. The identified Hammerstein models with parameter values $\{\hat{\mathbf{h}}, \hat{\mathbf{c}}\}$ for both loudspeakers are shown in Figure 3.6, with Figure 3.6(a) depicting the identified memoryless nonlinearity and Figure 3.6(b) depicting the identified FIR filter frequency response. From Figure 3.6(a), we observe that both loudspeakers possess a non-invertible smooth memoryless nonlinearity (class II). The memoryless nonlinearity of Loudspeaker 2 is seen to deviate from $g(x) = x$ over a wider amplitude input range than the memoryless nonlinearity of Loudspeaker 1. From Figure 3.6(b), we observe that for both loudspeakers, the identified FIR filter attenuates low frequencies, this effect being more pronounced for Loudspeaker 2.

Each audio signal in the test database (see Table 3.1) was played back by each of the two loudspeakers, once with and once without performing precompensation. The following settings were used in the precompensation using Algorithm 4:

- Identified Hammerstein model parameters $\mathbf{h}_m = \hat{\mathbf{h}}$, $\mathbf{c} = \hat{\mathbf{c}}$ are used.
- $N = 512$, $\alpha = 0.01$, $\beta = 0.1$, $\eta = 0.6$, and $K = 200$.

The simulations were performed for three distinct playback levels of the input signals, with relative levels $a = \{0.50, 0.75, 1.00\}$. The objective audio qual-



(a) (b)

Figure 3.6: Identified Hammerstein models for Loudspeaker 1 (solid lines) and Loudspeaker 2 (dashed lines): (a) memoryless nonlinearity (b) FIR filter frequency response.

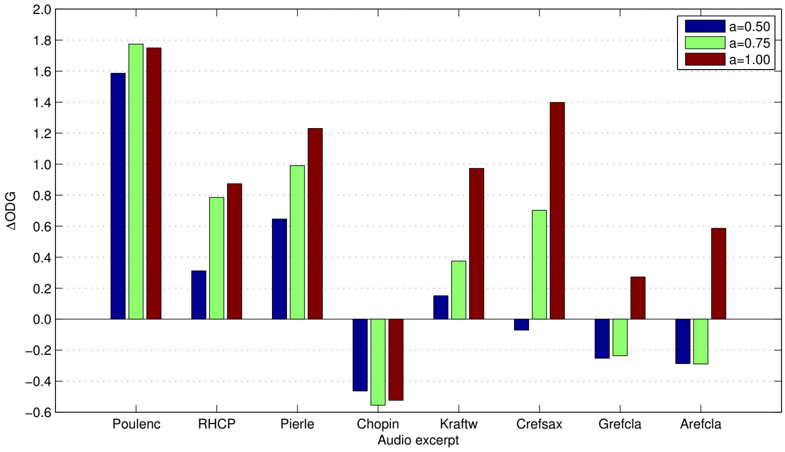
ity improvement for each audio signal was assessed by computing the ΔODG measure,

$$\Delta\text{ODG} = \text{ODG}(\mathbf{x}, \mathbf{y}^*) - \text{ODG}(\mathbf{x}, \mathbf{y}) \tag{3.45}$$

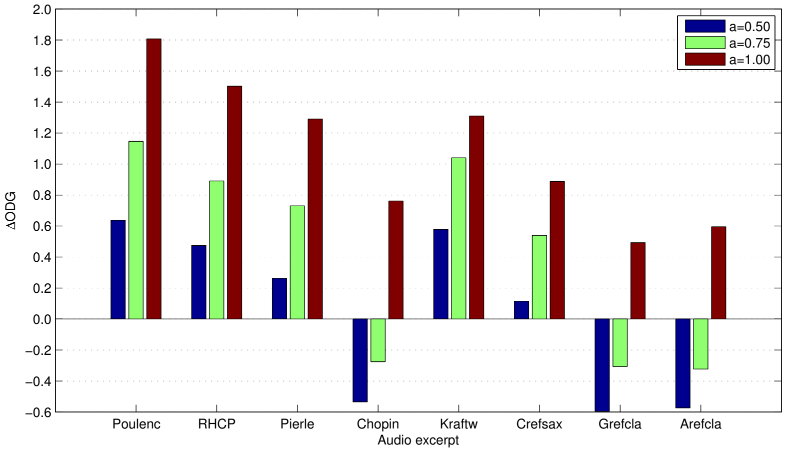
where $\mathbf{x}$ is the input signal, $\mathbf{y}$ is the loudspeaker output signal when no precompensation is applied, and $\mathbf{y}^*$ is the loudspeaker output signal when precompensation is applied.

The resulting ΔODG scores for Loudspeaker 1 and Loudspeaker 2 are shown in Figure 3.7(a) and Figure 3.7(b), respectively. For Loudspeaker 1 in Figure 3.7(a), positive audio quality improvement scores are observed for most audio excerpts. Increasing audio quality improvement scores are observed for increasing playback levels. The negative improvement scores for audio excerps Chopin (all playback levels), Grefcla and Arefcla (low playback levels) seem to indicate that there is a signal-specific threshold on the loudspeaker playback level below which performing precompensation is unnecessary and can even harm the resulting audio quality.

For Loudspeaker 2 in Figure 3.7(b), a similar pattern is observed with audio quality improvement scores that are increasing with the playback level. For the highest playback level $a = 1.00$, all precompensated audio excerpts result in an improved audio quality score. For the lower playback level, positive audio quality improvement scores are observed, except for the audio excerpts Chopin, Grefcla and Arefcla mentioned previously.



(a)



(b)

Figure 3.7: Precompensation of two loudspeakers using Hammerstein loudspeaker models - objective audio quality improvement scores for different audio excerpts, at average amplitude levels $= \{0.50, 0.75, 1.00\}$: (a) Loudspeaker 1 (b) Loudspeaker 2.

3.4.3 Identification of Hammerstein Model Parameters

The purpose of identification is to estimate the unknown Hammerstein model parameters $\mathbf{h}_L = [h_1, ..., h_L]^T$ (excluding h_0) and $\mathbf{c} = [c_1, ..., c_P]^T$ based on an observed output signal $\mathbf{y} = [y_1, ..., y_N]^T$ and an observed input signal $\mathbf{x} = [x_1, ..., x_N]^T$. In our experimental set-up, we have used a test audio signal having 30 seconds duration as the loudspeaker input signal, and measured the loudspeaker output signal in a soundproof room using a microphone placed on the axis of the loudspeaker at a distance of 10 centimeter. The estimation is done by solving the following optimization problem,

$$\begin{aligned} \{\hat{\mathbf{h}}_L, \hat{\mathbf{c}}\} &= \arg\min_{\mathbf{h}_L, \mathbf{c}} \; f_2(\mathbf{h}_L, \mathbf{c}) \\ &= \arg\min_{\mathbf{h}_L, \mathbf{c}} \; \frac{1}{N} \sum_{n=1}^{N} \left(y_n - \mathbf{h}_L^T \, \boldsymbol{\Psi}(\mathbf{x}_n) \, \mathbf{c} \right)^2 \end{aligned} \tag{3.46}$$

where $\mathbf{x}_n = [x_{n-1}, ..., x_{n-L}]^T$ and consequently $\boldsymbol{\Psi}(\mathbf{x}_n) = [\boldsymbol{\psi}(x_{n-1}), ..., \boldsymbol{\psi}(x_{n-L})]^T$. Given an initial estimate $\hat{\mathbf{h}}_L^0 \neq \mathbf{0}$, the normalized iterative algorithm [23] solves optimization problem (3.46) by performing the following steps in each iteration k:

1. An estimate $\bar{\mathbf{c}}^k$ is computed as the solution of the least-squares problem (3.46) for fixed $\mathbf{h}_L = \hat{\mathbf{h}}_L^{k-1}$,

$$\bar{\mathbf{c}}^k = \arg\min_{\mathbf{c} \in \mathbb{R}^P} \; f_2(\hat{\mathbf{h}}_L^{k-1}, \mathbf{c}). \tag{3.47}$$

2. An estimate $\bar{\mathbf{h}}_L^k$ is computed as the solution of the least-squares problem (3.46) for fixed $\mathbf{c} = \bar{\mathbf{c}}^k$,

$$\bar{\mathbf{h}}_L^k = \arg\min_{\mathbf{h}_L \in \mathbb{R}^L} \; f_2(\mathbf{h}_L, \bar{\mathbf{c}}^k). \tag{3.48}$$

3. A normalization is performed so that $||\hat{\mathbf{h}}_L^k|| = 1$,

$$\hat{\mathbf{h}}_L^k = \frac{\xi^k \bar{\mathbf{h}}_L^k}{||\bar{\mathbf{h}}_L^k||} \tag{3.49}$$

$$\hat{\mathbf{c}}^k = \xi^k \bar{\mathbf{c}}^k ||\bar{\mathbf{h}}_L^k|| \tag{3.50}$$

where $\xi^k = \pm 1$ is the sign of the first non-zero entry of $\bar{\mathbf{h}}_L^k$.

3.5 Conclusions

In this paper, an embedded-optimization-based loudspeaker precompensation algorithm using a Hammerstein loudspeaker model has been presented. The loudspeaker precompensation consists in a per-frame signal optimization. In order to minimize the perceptible distortion incurred in the loudspeaker, a percepually meaningful optimization criterion has been constructed. Depending on the invertibility and the smoothness of the memoryless nonlinearity, different first order optimization methods have been proposed for solving the per-frame optimization problems. Objective evaluation experiments using synthetic and identified loudspeaker models have shown that the proposed loudspeaker precompensation algorithm provides a significant audio quality improvement, especially so at high playback levels.

Bibliography

[1] K. Lashkari, “A modified Volterra-Wiener-Hammerstein model for loudspeaker precompensation,” in *Proc. 39th Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, Oct. 2005, pp. 344–348.

[2] C.-T. Tan, B. C. J. Moore, and N. Zacharov, “The effect of nonlinear distortion on the perceived quality of music and speech signals,” *J. Audio Eng. Soc.*, vol. 51, no. 11, pp. 1012–1031, Nov. 2003.

[3] B. C. Moore, C.-T. Tan, N. Zacharov, and V.-V. Mattila, “Measuring and predicting the perceived quality of music and speech subjected to combined linear and nonlinear distortion,” *Journal of the Audio Engineering Society*, vol. 52, no. 12, pp. 1228–1244, 2004.

[4] M. Karjalainen and T. Paatero, “Equalization of loudspeaker and room responses using Kautz filters: Direct least squares design,” *EURASIP Journal on Advances in Signal Processing*, vol. 2007, pp. 1–13, 2007.

[5] M. V. K. Shi, G. T. Zhou, “Compensation for nonlinearity in a Hammerstein system using the coherence function with application to nonlinear acoustic echo cancellation,” *IEEE Transactions on Signal processing*, vol. 55, no. 12, pp. 5853–5858, 2007.

[6] B. Defraene, T. van Waterschoot, M. Diehl, and M. Moonen, “Perception-based nonlinear loudspeaker compensation through embedded convex optimization,” in *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO), Bucharest, Romania*, Aug. 2012, pp. 2472 –2476.

[7] ——, “Embedded-optimization-based loudspeaker compensation using a generic Hammerstein loudspeaker model,” in *2013 Proceedings of the 21st*

European Signal Processing Conference (EUSIPCO), Marrakech, Morocco, Sept. 2013.

[8] B. Defraene, T. van Waterschoot, H. J. Ferreau, M. Diehl, and M. Moonen, "Real-time perception-based clipping of audio signals using convex optimization," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 10, pp. 2657 –2671, Dec. 2012.

[9] E. Terhardt, "Calculating virtual pitch," *Hearing Res.*, vol. 1, no. 2, pp. 155 – 182, 1979.

[10] T. Painter and A. Spanias, "Perceptual coding of digital audio," *Proc. IEEE*, vol. 88, no. 4, pp. 451–515, Apr. 2000.

[11] ISO/IEC, "11172-3 Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s - Part 3: Audio," 1993.

[12] E. Zwicker and H. Fastl, *Psychoacoustics: facts and models*, 2nd ed. Springer, 1999.

[13] L. Armijo, "Minimization of functions having lipschitz continuous first partial derivatives." *Pacific Journal of mathematics*, vol. 16, no. 1, pp. 1–3, 1966.

[14] Y. Nesterov, *Introductory lectures on convex optimization.* Springer, 2004.

[15] B. Defraene, T. van Waterschoot, M. Diehl, and M. Moonen, "A fast projected gradient optimization method for real-time perception-based clipping of audio signals," in *Proc. ICASSP 2011*, Prague, Czech Republic, May 2011, pp. 333–336.

[16] A. Ostrowski, "A quantitative formulation of Sylvester's law of inertia," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 45, no. 5, p. 740, 1959.

[17] F. Poulenc, "Sonata for flute and piano (Cantilena)," French Flute Music, Naxos 8.557328, 2005.

[18] Red Hot Chili Peppers, "Californication," Californication, Warner Bros. Records 9362473862, 1999.

[19] An Pierlé & White Velvet, "Mexico," An Pierlé & White Velvet, PIAS Recordings 941.0170.020, 2006.

[20] F. Chopin, "Waltz op. 69 no. 2," Favourite Piano Works (Vladimir Ashkenazy), Decca 02894448302, 1995.

[21] Kraftwerk, "Tour de France Etape 1," Minimum-Maximum, EMI Music 724356061620, 2005.

[22] International Telecommunications Union Recommendation BS.1387, "Method for objective measurements of perceived audio quality," 1998.

[23] E.-W. Bai and D. Li, "Convergence of the iterative Hammerstein system identification algorithm," *Automatic Control, IEEE Transactions on*, vol. 49, no. 11, pp. 1929–1940, 2004.

Chapter 4

Subjective Audio Quality Evaluation

4.1 Introduction

In Chapter 2 and Chapter 3, novel embedded-optimization-based precompensation algorithms have been presented which incorporate a perceptual model. In Chapter 2, a perception-based clipping algorithm has been presented using embedded convex optimization. In Chapter 3, a loudspeaker precompensation algorithm has been presented using embedded optimization. Both algorithms have been evaluated using objective measures of audio quality, such as PEAQ [1] and Rnonlin [2], for which significant improvements have been reported. In Subsection 2.4, it has been concluded from evaluation experiments that the perception-based clipping algorithm results in significantly higher objective audio quality scores than existing clipping techniques. In Subsection 3.4, it has been concluded from experiments using synthetic and identified loudspeaker models that the proposed loudspeaker precompensation algorithm provides a significant objective audio quality improvement, especially at high playback levels.

The objective measures used in the evaluation experiments are predictive models of the perceived audio quality. Taking a reference signal (i.e. the clean signal) and a signal under test (i.e. the processed signal) as an input, the objective measure of perceived audio quality is calculated through sequential application of a human auditory model and a cognitive model. Resultingly, a perceived audio quality score is attributed to the processed signal with respect to the clean signal.

Although considerable research efforts have been invested in the development and standardization of such objective audio quality measures, limits on their accuracy and applicability are inevitable. Firstly, a given objective audio quality measure is applicable to assess the perceptibility of the particular class of distortions it was designed for, and care should be taken when applying the measure to assess the perceptibility of other distortions. For example, PEAQ was designed to assess the perceptibility of degradations commonly encountered in audio codecs. Secondly, the current objective audio quality measures are predictive models with a limited accuracy, and as a result cannot replace the subjective evaluation of human listeners [3].

For these reasons, it is necessary to perform a formal subjective evaluation to accurately measure the perceived audio quality for the algorithms presented in Chapter 2 and Chapter 3. The subjective evaluation takes the form of a formal listening test, in which it is assessed how listeners perceive an audio signal by asking them to quantify their experience.

The aim of our formal listening test is threefold:

1. To assess the subjective perceived audio quality of the perception-based clipping algorithm and the loudspeaker precompensation algorithm.
2. To assess the subjective perceived audio quality of audio signals subject to combined linear and nonlinear distortions, as compared to the case where only nonlinear distortions are present.
3. To assess the correlation between the objective and subjective perceived audio quality scores.

This chapter is organized as follows. In Section 4.2, the research questions to be answered through the listening test are defined and the corresponding hypotheses are formulated. In Section 4.3, the experimental design and set-up of the listening test is discussed. In Section 4.4, the test results are reported and the formulated hypotheses are statistically tested. In Section 4.5, some concluding remarks are presented.

4.2 Research Questions and Hypotheses

The two main research questions to be answered through performing a formal subjective listening test relate to the perceived audio quality of the proposed perception-based clipping and loudspeaker precompensation algorithms. These two research questions are formulated as follows:

Question 1 How does the perceived audio quality of audio signals clipped by the proposed perception-based clipping algorithm compare to that of audio signals clipped by the hard clipping algorithm?

Question 2 How does the perceived audio quality of audio signals processed by the proposed loudspeaker precompensation algorithm before being fed into the Hammerstein loudspeaker model compare to that of audio signals not processed before being fed into the Hammerstein loudspeaker model?

A third research question relates to the perceived audio quality of audio signals subject to a combined linear and nonlinear distortion, as compared to audio signals subject to only nonlinear distortion. This third research question is formulated as follows:

Question 3 How does the perceived audio quality of audio signals subject to a certain nonlinear hard clipping distortion compare to that of audio signals subject to a combination of the same nonlinear hard clipping distortion and an additional linear distortion?

A fourth research question relates to the correlation between the objective and subjective perceived audio quality scores.

Question 4 How do the subjective perceived audio quality scores correlate with the objective perceived audio quality scores?

The corresponding research hypotheses, that may or may not be rejected, are formulated as follows:

Hypothesis 1 The perceived audio quality of audio signals clipped by the perception-based clipping algorithm is identical to that of audio signals clipped by the hard clipping algorithm.

Hypothesis 2 The perceived audio quality of audio signals processed by the proposed loudspeaker precompensation algorithm before being fed into the Hammerstein loudspeaker model is identical to that of audio signals not processed before being fed into the Hammerstein loudspeaker model.

Hypothesis 3 The perceived audio quality of audio signals subject to a certain nonlinear hard clipping distortion is identical to that of audio signals subject to a combination of the same nonlinear hard clipping distortion and an additional linear distortion.

Hypothesis 4 There is no correlation between subjective perceived audio quality scores and objective perceived audio quality scores.

Table 4.1: Audio excerpts used for subjective audio quality evaluation.

Nr.	Name	Texture	Style	Duration [s]	Origin
1	rhcp.wav	polyphonic	rock	9.8	[4]
2	chopin.wav	monophonic	classical	17.8	[5]
3	poulenc.wav	polyphonic	classical	17.8	[6]
4	crefsax.wav	monophonic	classical	10.9	[1]

4.3 Experimental Design and Set-up

A representative sample of 19 test subjects having considerable musical listening and performance experience was selected to perform the listening test. All subjects were remunerated for their participation.

The stimuli presented to the test subjects consisted of four audio excerpts (detailed in Table 4.1), each of which were presented in 12 different processing scenarios:

- Processing scenarios S_1-S_3: Hard symmetrical clipping as in (2.33), where the clipping level U is selected such that the processed audio signal has a PEAQ ODG of -1, -2 and -3, for the respective processing scenarios S_1, S_2 and S_3.
- Processing scenarios S_4-S_6: Perception-based clipping as in Algorithm 3 described in Subsection 2.3.4, with parameter values $N = 512$, $P = 128$, $\alpha = 0.04$, and the same clipping level U as used in the respective processing scenarios S_1, S_2 and S_3.
- Processing scenarios S_7-S_9: Uncompensated Hammerstein loudspeaker model consisting of:
  - Hard clipping memoryless nonlinearity (class III) with the same clipping level U as used in the respective processing scenarios S_1, S_2 and S_3.
  - Linear FIR filter ($L = 128$) with impulse response $h[n]$ designed using the frequency sampling method `fir2` in Matlab, having a required magnitude response $[1, 0.95, 0.75, 0.50, 0.20, 0]^T$ for the frequencies $[0, 0.2, 0.4, 0.6, 0.8, 1]^T \times f_{\text{Nyquist}}$.
- Processing scenarios S_{10}-S_{12}: Precompensated Hammerstein loudspeaker model, using the same Hammerstein model settings as in the respective processing scenarios S_7, S_8 and S_9, and using the following settings in the precompensation: Algorithm 5 described in Subsection 3.3.4, with $N = 512$, $\alpha = 0.01$, $\gamma_m^0 = \sqrt{\frac{\mu_m}{C_m}}$, $K = 500$.

This resulted in a total of $N_{\text{ps}} = 4\times12=48$ pairs of stimuli (each consisting of the original unprocessed audio signal and the corresponding processed audio signal) that were presented to the test subjects. For each pair of stimuli, the

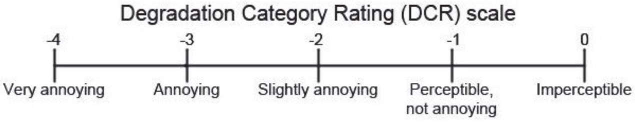
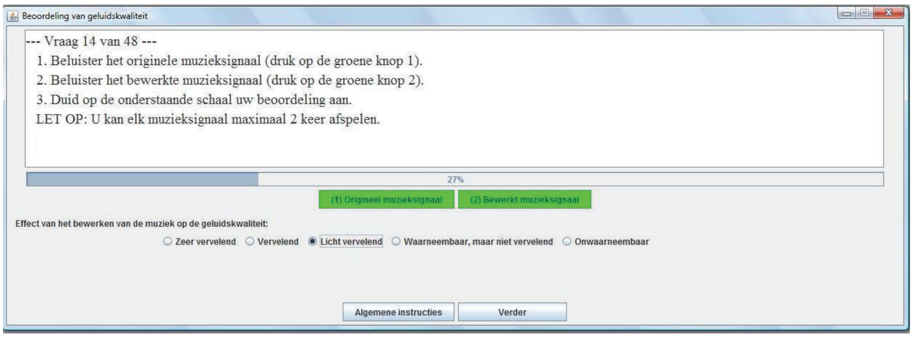

Degradation Category Rating (DCR) scale
-4
-3
-2
-1
0
Very annoying
Annoying
Slightly annoying
Perceptible, not annoying
Imperceptible

Figure 4.1: ITU-T Degradation Category Rating (DCR) scale (adapted from [3]).


Beoordeling van geluidskwaliteit
27%
(1) Origineel muzieksignaal
(2) Bewerkt muzieksignaal
Algemene instructies
Verder

--- Vraag 14 van 48 ---

1. Beluister het originele muzieksignaal (druk op de groene knop 1).
2. Beluister het bewerkte muzieksignaal (druk op de groene knop 2).
3. Duid op de onderstaande schaal uw beoordeling aan.

LET OP: U kan elk muzieksignaal maximaal 2 keer afspelen.

Effect van het bewerken van de muziek op de geluidskwaliteit:

○ Zeer vervelend ○ Vervelend ● Licht vervelend ○ Waarneembaar, maar niet vervelend ○ Onwaarneembaar

Figure 4.2: Graphical User Interface (in Dutch) of listening test software for stimulus presentation and response collection.

test subjects were asked to rate the perceived audio quality degradation of the presented processed signal with the original audio signal as a reference, using the ITU-T Degradation Category Rating (DCR) [7] scale depicted in Figure 4.1.

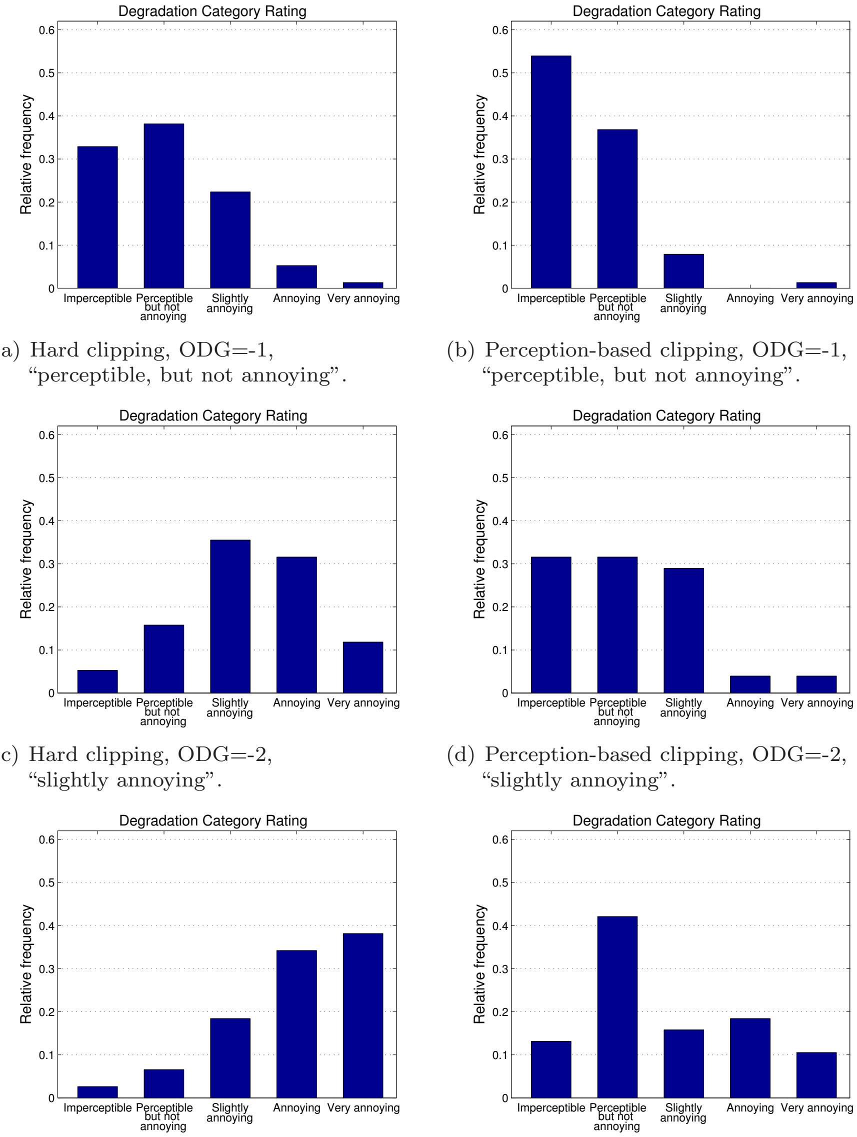
The listening tests were performed in a soundproof and well-illuminated test room. Stimuli were presented to the test subjects through high-quality circumaural headphones[1] connected to a soundcard-equipped laptop[2]. Self-developed software was used to automate stimulus presentation and response collection (see Figure 4.2). The playback level was fixed at a comfortable level.

Prior to the listening test, the subjects were provided with written instructions, which were verbally reviewed by the experimenter. Before the first pair of stimuli was presented, the subjects were familiarized with the effects of linear and nonlinear distortion on audio signals, by successively listening to an original sample audio signal and its distorted version. The presentation order of the pairs of stimuli was randomized using an altered Latin square scheme [3], thus eliminating possible bias effects due to order effects and sequential

[1] Sennheiser HD 439: dynamic, closed transducer, frequency response 17-22500Hz, Sound Pressure Level 112 dB, Total Harmonic Distortion < 0.1%.

[2] Sony Vaio VGN-CR41: Intel Core 2 duo T5550 processor @1.83Ghz, 3GB RAM, Realtek sound card, Intel GMA X3100 Graphics Processor.

dependencies.



(a) Hard clipping, ODG=-1, "perceptible, but not annoying".

(b) Perception-based clipping, ODG=-1, "perceptible, but not annoying".

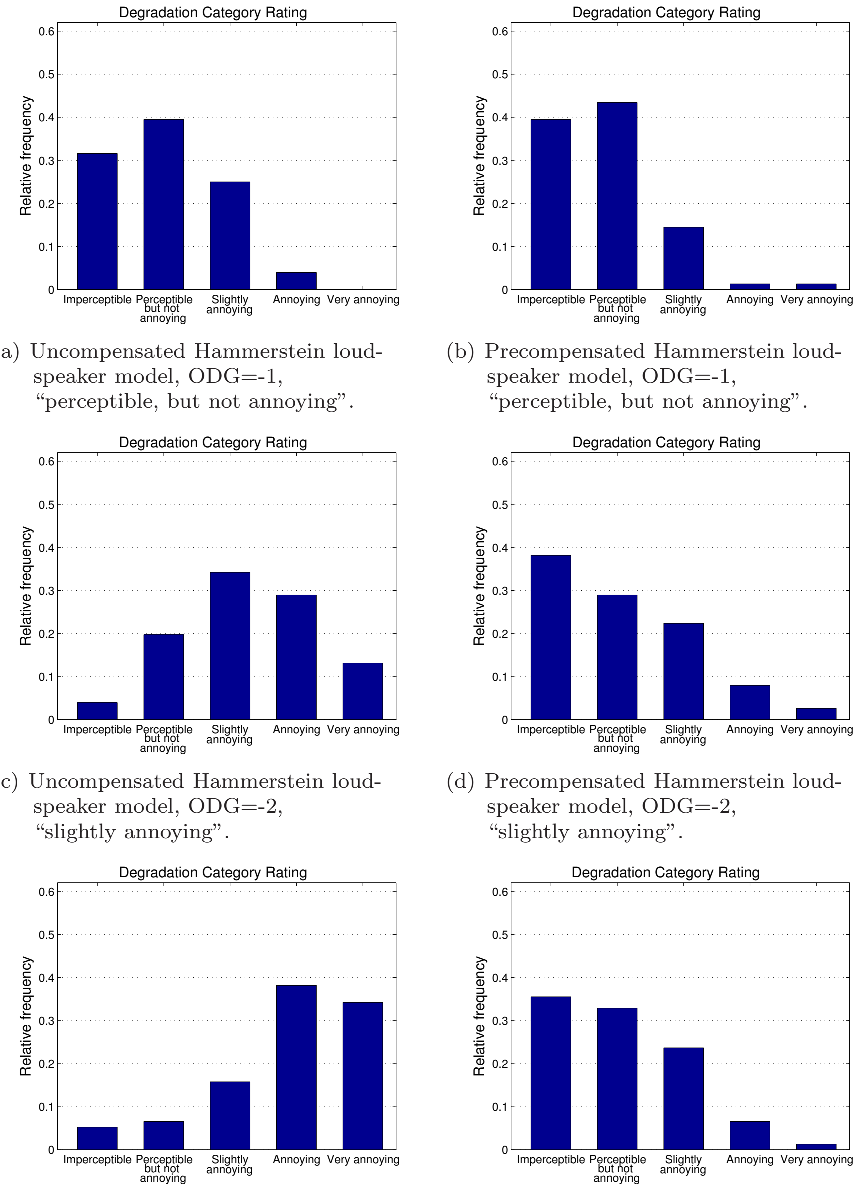
(c) Hard clipping, ODG=-2, "slightly annoying".

(d) Perception-based clipping, ODG=-2, "slightly annoying".

(e) Hard clipping, ODG=-3, "annoying".

(f) Perception-based clipping, ODG=-3, "annoying".

Figure 4.3: Histograms of DCR responses for the hard clipping algorithm (left) and the perception-based algorithm (right), for different hard clipped input ODGs.



(a) Uncompensated Hammerstein loudspeaker model, ODG=-1, "perceptible, but not annoying".

(b) Precompensated Hammerstein loudspeaker model, ODG=-1, "perceptible, but not annoying".

(c) Uncompensated Hammerstein loudspeaker model, ODG=-2, "slightly annoying".

(d) Precompensated Hammerstein loudspeaker model, ODG=-2, "slightly annoying".

(e) Uncompensated Hammerstein loudspeaker model, ODG=-3, "annoying".

(f) Precompensated Hammerstein loudspeaker model, ODG=-3, "annoying".

Figure 4.4: Histograms of DCR responses for audio signals without loudspeaker precompensation (left) and with loudspeaker precompensation (right), for different hard clipped input ODGs.

4.4 Results and Statistical Analysis

The raw results of the listening test, which had an average duration of 35 minutes per test subject, are presented in subsection 4.4.1, after which the hypotheses formulated in 4.2 are statistically tested in subsections 4.4.2 and 4.4.3.

4.4.1 Test Subject Responses

The raw data resulting from the listening test consists of a categorical DCR response by each of the 19 test subjects, for each of the 48 presented pairs of stimuli.

Figure 4.3 shows histograms of the obtained DCR responses for the hard clipping algorithm and for the perception-based clipping algorithm, and this for the three considered clipping levels, corresponding to PEAQ ODG levels {-1,-2,-3}. It is observed that the response histograms for the perception-based clipping algorithm have a higher probability mass in the two leftmost bins compared to the response histograms for the hard clipping algorithm. This indicates that overall, the test subjects have rated the perceived audio quality for the perception-based clipping algorithm more favorably than the hard clipping algorithm.

Figure 4.4 shows histograms of the obtained DCR responses for audio signals without Hammerstein loudspeaker model precompensation and for audio signals with Hammerstein loudspeaker model precompensation, again for the three considered clipping levels, corresponding to PEAQ ODG levels {-1,-2,-3}. It is observed that the response histograms for the audio signals with Hammerstein model precompensation have a higher probability mass in the two leftmost bins compared to the response histograms for the audio signals without Hammerstein model precompensation. This indicates that overall, the test subjects have rated the perceived audio quality for the audio signals with Hammerstein model precompensation more favorably than the audio signals without Hammerstein model precompensation.

4.4.2 Statistical Hypothesis Testing

The categorical DCR responses were first converted to integers according to the scale in Figure 4.1. The following statistical analysis was performed on the obtained numerical set of DCR responses.

Testing Hypothesis 1

Let us denote the population DCR responses corresponding to audio signals processed by the hard clipping algorithm and the perception-based clipping algorithm by random variables R_{HC} and R_{PBC}, respectively.

Based on the sample DCR responses, we tested the following statistical hypothesis H_0^1 against its alternative H_a^1:

$$\begin{aligned} H_0^1 &: \tilde{R}_{\mathrm{HC}} = \tilde{R}_{\mathrm{PBC}} && (4.1) \\ H_a^1 &: \tilde{R}_{\mathrm{HC}} \leq \tilde{R}_{\mathrm{PBC}} && (4.2) \end{aligned}$$

where $\tilde{R}$ is the population median of the random variable R.

This statistical hypothesis was tested for all three considered ODGs using one-tailed Wilcoxon-Mann-Whitney tests [8] with significance level $\alpha = 0.05$. The resulting one-sided P-values are synthesized in the first column of Table 4.2.

From the obtained P-values, we conclude that the null hypothesis (4.1) can be rejected in favor of the alternative (4.2) for all considered ODGs. The perceived audio quality of audio signals clipped by the perception-based clipping algorithm is not identical but significantly superior to that of audio signals clipped by the hard clipping algorithm, and this for all considered ODGs.

Testing Hypothesis 2

Let us denote the population DCR responses corresponding to audio signals processed by the uncompensated Hammerstein loudspeaker model and the precompensated Hammerstein loudspeaker model by random variables R_{UNCOMP} and R_{COMP}, respectively.

Based on the sample DCR responses, we tested the following statistical hypothesis H_0^2 against its alternative H_a^2:

$$\begin{aligned} H_0^2 &: \tilde{R}_{\mathrm{UNCOMP}} = \tilde{R}_{\mathrm{COMP}} && (4.3) \\ H_a^2 &: \tilde{R}_{\mathrm{UNCOMP}} \leq \tilde{R}_{\mathrm{COMP}}. && (4.4) \end{aligned}$$

This statistical hypothesis was tested for all three considered ODGs using one-tailed Wilcoxon-Mann-Whitney tests [8] with significance level $\alpha = 0.05$. The resulting one-sided P-values are synthesized in the second column of Table 4.2.

From the obtained P-values, we conclude that the null hypothesis (4.3) can be rejected in favor of the alternative (4.4) for ODGs of -2 and -3. The perceived audio quality of audio signals processed by the proposed loudspeaker precompensation algorithm before being fed into the Hammerstein loudspeaker model is not identical, but significantly superior to that of audio signals not pro-

Table 4.2: P-values from one-tailed Wilcoxon-Mann-Whitney tests on sample DCR responses. Significant P-values with respect to $\alpha = 0.05$ in bold.

Null hypothesis →	H_0^1	H_0^2	H_0^3
ODG=-1	**0.0006**	0.0616	0.5062
ODG=-2	<**0.0001**	<**0.0001**	0.4398
ODG=-3	<**0.0001**	<**0.0001**	0.3374

cessed before being fed into the Hammerstein loudspeaker model, and this for moderate to high amplitude levels.

Testing Hypothesis 3

Based on the sample DCR responses, we tested the following statistical hypothesis H_0^3 against its alternative H_a^3:

$$H_0^3 \ : \ \tilde{R}_{\mathrm{UNCOMP}} = \tilde{R}_{\mathrm{HC}} \tag{4.5}$$

$$H_a^3 \ : \ \tilde{R}_{\mathrm{UNCOMP}} \leq \tilde{R}_{\mathrm{HC}}. \tag{4.6}$$

This statistical hypothesis was tested for all three considered ODGs using one-tailed Wilcoxon-Mann-Whitney tests [8] with significance level $\alpha = 0.05$. The resulting one-sided P-values are synthesized in the third column of Table 4.2.

From the obtained P-values, we conclude that the null hypothesis (4.5) cannot be rejected in favor of the alternative (4.6) for any ODG. This means that there is no reason to believe that the perceived audio quality of audio signals subject to a certain nonlinear hard clipping distortion would be superior to that of audio signals subject to a combination of the same nonlinear hard clipping distortion and an additional linear distortion.

4.4.3 Correlation Between Subjective and Objective Scores

As mentioned in Section 4.1, the PEAQ ODG measure has been designed to objectively assess the perceptibility of degradations commonly encountered in audio codecs. However, the nature of signal distortions introduced by clipping algorithms and (un)compensated Hammerstein loudspeaker models can be rather different as compared to signal distortions introduced by audio codecs. Therefore, we will investigate the validity of using PEAQ ODG as an objective audio quality measure in these alternative scenarios.

The correlation between subjective and objective scores is the most obvious criterion to validate an objective method. Let us denominate the mean DCR responses over all 19 test subjects as MDCR responses. Then we can calculate the sample Pearson correlation coefficient $\hat{\rho}$ between the subjective MDCR responses and the objective ODG scores as follows,

$$\hat{\rho} = \frac{\sum_{i=1}^{N_{\mathrm{ps}}}(\mathrm{MDCR}_i - \overline{\mathrm{MDCR}})(\mathrm{ODG}_i - \overline{\mathrm{ODG}})}{\sqrt{\sum_{i=1}^{N_{\mathrm{ps}}}(\mathrm{MDCR}_i - \overline{\mathrm{MDCR}})^2}\sqrt{\sum_{i=1}^{N_{\mathrm{ps}}}(\mathrm{ODG}_i - \overline{\mathrm{ODG}})^2}} \tag{4.7}$$

where

$$\overline{\mathrm{MDCR}} = \sum_{i=1}^{N_{\mathrm{ps}}} \mathrm{MDCR}_i \tag{4.8}$$

and

$$\overline{\mathrm{ODG}} = \sum_{i=1}^{N_{\mathrm{ps}}} \mathrm{ODG}_i. \tag{4.9}$$

Testing Hypothesis 4

The scatter plot of the mean subjective DCR scores against the objective ODG scores is shown in Figure 4.5. Based on the resulting sample Pearson correlation coefficient value $\hat{\rho} = 0.67$, we tested the following statistical hypothesis H_0^4 against its alternative H_a^4:

$$H_0^4 \ : \ \rho = 0 \tag{4.10}$$

$$H_a^4 \ : \ \rho > 0. \tag{4.11}$$

where ρ is the population Pearson correlation coefficient.

This statistical hypothesis was tested [9] with significance level $\alpha = 0.05$ by using a one-tailed t-test having $N_{\mathrm{ps}} - 2$ degrees of freedom for the test statistic value $t = |\hat{\rho}|\frac{\sqrt{N_{\mathrm{ps}}-2}}{\sqrt{1-\hat{\rho}^2}}$.

The resulting one-sided P-value is $1.206 \cdot 10^{-7} < \alpha$, which means that the null hypothesis (4.10) can be confidently rejected in favor of the alternative (4.11). In conclusion, there is a significant positive correlation between subjective audio quality scores and objective PEAQ ODG scores, supporting the validity of using PEAQ ODG as an objective audio quality measure for audio signals processed by clipping algorithms and (un)compensated Hammerstein loudspeaker models.

4.5 Conclusions

Because of the limited applicability and accuracy of objective audio quality measures, a formal subjective listening test has been performed in order to as-

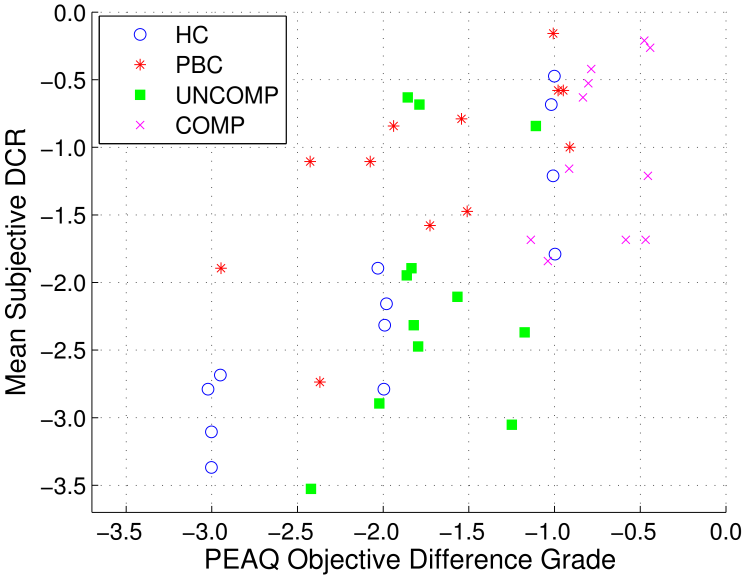


Figure 4.5: Scatter plot of obtained objective ODG scores vs. mean subjective DCR scores.

sess the perceived audio quality of audio signals processed by the proposed perception-based clipping and loudspeaker precompensation algorithms. A representative sample of 19 test subjects provided subjective audio quality responses to 48 pairs of stimuli. Statistical hypothesis tests on the obtained responses led to the following conclusions:

1. The perceived audio quality of audio signals clipped by the perception-based clipping algorithm is significantly superior to that of audio signals clipped by the hard clipping algorithm, and this for all considered ODGs.
2. The perceived audio quality of audio signals processed by the proposed loudspeaker precompensation algorithm before being fed into the Hammerstein loudspeaker model is significantly superior to that of audio signals not processed before being fed into the Hammerstein loudspeaker model, and this for moderate to high amplitude levels.
3. There is no reason to believe that the perceived audio quality of audio signals subject to a certain nonlinear hard clipping distortion would be superior to that of audio signals subject to a combination of the same nonlinear hard clipping distortion and an additional linear distortion.
4. There is a significant positive correlation between subjective audio quality scores and objective PEAQ ODG scores.

Bibliography

[1] International Telecommunications Union Recommendation BS.1387, “Method for objective measurements of perceived audio quality,” 1998.

[2] C.-T. Tan, B. C. J. Moore, N. Zacharov, and V.-V. Mattila, “Predicting the perceived quality of nonlinearly distorted music and speech signals,” *J. Audio Eng. Soc.*, vol. 52, no. 7–8, pp. 699–711, Jul. 2004.

[3] S. Bech and N. Zacharov, *Perceptual audio evaluation-Theory, method and application.* Wiley, 2007.

[4] Red Hot Chili Peppers, “Californication,” Californication, Warner Bros. Records 9362473862, 1999.

[5] F. Chopin, “Waltz op. 69 no. 2,” Favourite Piano Works (Vladimir Ashkenazy), Decca 02894448302, 1995.

[6] F. Poulenc, “Sonata for flute and piano (Cantilena),” French Flute Music, Naxos 8.557328, 2005.

[7] International Telecommunications Union Recommendation P. 800, “Methods for subjective determination of transmission quality,” 1996.

[8] F. Wilcoxon, “Individual comparisons by ranking methods,” *Biometrics bulletin*, vol. 1, no. 6, pp. 80–83, 1945.

[9] S. Siegel, *Nonparametric statistics for the behavioral sciences.* McGraw-Hill, 1956.

Chapter 5

Embedded Hardware Implementation

5.1 Introduction

This chapter considers the embedded hardware implementation on a field-programmable gate array (FPGA) of the core of the perception-based clipping algorithm presented in Chapter 2. As the clipping algorithm is intended for real-time audio signal processing on a typically small and portable device, the hardware design should preferably have a low latency, low power consumption, and low resource usage, while at the same time preserving a maximal audio quality.

The core of the perception-based clipping algorithm is formed by the solution of a per-frame convex optimization problem. This clearly represents the most computationally intensive part of the perception-based clipping algorithm, hence our focus to restrict the hardware implentation to the optimization problem solution method. The iterative optimal projected gradient method proposed in subsection 2.3.4 has several properties making it eligible for an efficient hardware implementation. Firstly, polynomial upper and lower bounds have been derived on the number of necessary iterations of the method to ensure a given solution accuracy. This allows to choose a priori a fixed number of iterations to ensure the desired solution accuracy, and at the same time allows to leave out a convergence test at every iteration. Secondly, the method does not require the solution of a linear system of equations at every itera-

The work presented in this chapter was carried out in close cooperation with Dr. Andrea Suardi, Department of Electrical and Electronic Engineering, Imperial College, London, UK.

tion (in contrast to many other convex optimization methods), which in many cases precludes a fast hardware implementation [1]. Lastly, both the projection and the gradient operation constituting an iteration of the method possess a structure that makes them amenable for efficient hardware implementation.

FPGA technology has been chosen to implement the optimal projected gradient method. FPGAs are programmable devices containing repeated fields of small logic blocks and elements, called configurable logic blocks (CLBs) or Logic Array Blocks (LABs) depending on the vendor. For digital signal processing purposes, FPGAs offer a number of advantages. Firstly, the reconfigurability of FPGAs provides a high design flexibility, without the need for any physical IC fabrication facility. Secondly, FPGAs can achieve a high computing power for DSP applications by taking advantage of hardware parallellism. Lastly, the growing availability of high-level software tools has decreased the FPGA design learning curve and offers valuable intellectual property (IP) cores (prebuilt functions) for advanced control and signal processing applications [2].

The proposed embedded hardware implementation on FPGA allows to investigate different trade-offs between the audio quality, latency, resource usage, and power consumption of the algorithm. The choice between fixed-point arithmetic and floating-point arithmetic as well as the selection of the corresponding word length form a crucial aspect in these trade-offs. The impact of these choices has been quantified.

This chapter is organized as follows. In Section 5.2, the optimal projected gradient method is reviewed and transformed into an FPGA implementation architecture. In Section 5.3, some FPGA implementation issues are discussed, concerning floating-point vs. fixed-point number representations and concerning the implementation of the Fast Fourier Transform. In Section 5.4, the results of bit-accurate simulations are presented and the resulting trade-offs are analyzed. In Section 5.5, some concluding remarks are presented.

5.2 Embedded Hardware Architecture

5.2.1 Optimal Projected Gradient Algorithm

Algorithm 6 shows the original optimal projected gradient algorithm used for efficiently solving the per-frame clipping optimization problem (omitting the frame index m here for simplicity),

$$\mathbf{y}^* = \arg\min_{\mathbf{y}\in\mathbb{R}^N} \frac{1}{2} (\mathbf{y}-\mathbf{x})^H \, \mathbf{D}^H\mathbf{W}\mathbf{D} \, (\mathbf{y}-\mathbf{x}) \quad \text{s.t.} \quad \mathbf{l} \leq \mathbf{y} \leq \mathbf{u} \tag{5.1}$$

where $\mathbf{x}$ is the input frame, $\mathbf{y}$ is the output frame, $\mathbf{D} \in \mathbb{C}^{N\times N}$ is the unitary DFT matrix, $\mathbf{W} = \text{diag}(\mathbf{w})$ is a diagonal weighting matrix of the perceptual

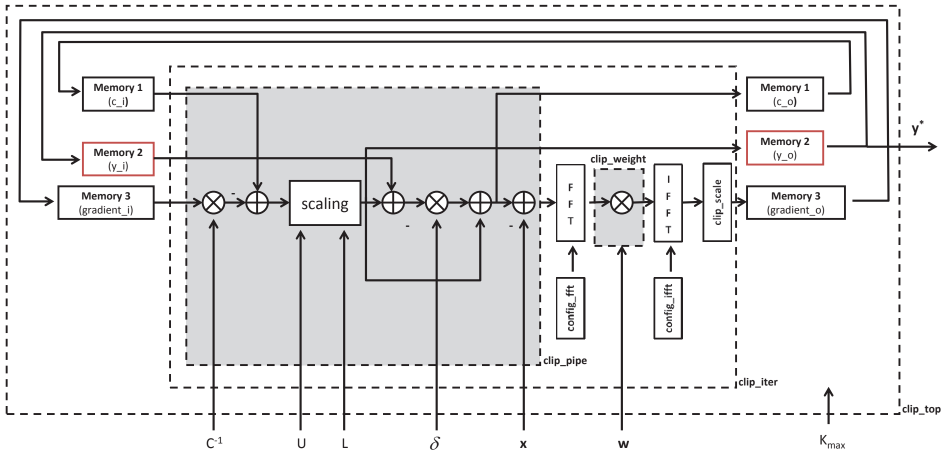


Figure 5.1: Block diagram representing the implementation of the clipping optimal projected gradient algorithm onto FPGA.

weights $\mathbf{w}$, and $\mathbf{u} = U\mathbb{1}_N$ and $\mathbf{l} = L\mathbb{1}_N$ contain the upper and lower clipping levels.

In order to allow for an efficient hardware implementation, we have performed the following organizational changes in Algorithm 6 without altering its functionality, as shown in the adapted Algorithm 7:

1. Several algorithm parameters need not be computed on-line, but can rather be precomputed off-line:
   - The per-iteration weights δ^k are precomputed off-line, using lines 2-3-4-10-11 of Algorithm 6. The resulting precomputed weight vector $\boldsymbol{\delta} = [\delta^0, \delta^1, \ldots, \delta^{K_{\max}-1}]^T$ is defined as an input to the adapted Algorithm 7.
   - The Lipschitz constant C is precomputed off-line, and the inverse of the Lipschitz constant C^{-1} is defined as an input to the adapted Algorithm 7. This way, the division operation by C on line 8 of Algorithm 6, is replaced by a more hardware-friendly multiplication with C^{-1} on line 5 of the adapted Algorithm 7.
2. The computation of the gradient was rescheduled from the start of the iteration (line 7 in Algorithm 6) to the end of the iteration (line 8 in Algorithm 7), thus reducing the overall algorithm delay.

5.2.2 FPGA Implementation Architecture

The optimal projected gradient algorithm presented in Algorithm 7 for solving the per-frame clipping optimization problem, has been implemented onto FPGA using a high level synthesis tool (Vivado HLS) and C language. This approach was chosen in order to facilitate the exploration of design trade-offs and to speed up design verification time, compared to a classical approach based on hardware description languages (e.g. VHDL or Verilog).

Algorithm 6 Original optimal projected gradient algorithm

```
Input $\mathbf{x} \in \mathbb{R}^N$, $\mathbf{w} \in \mathbb{R}^N$, $K_{\max}$, $L$, $U$, $\gamma^0 \in (0,1)$
Output $\mathbf{y}^* \in \mathbb{R}^N$
$\mathbf{y}^0 = \mathbf{c}^0 = \mathbf{x}$
$C = \max_{0 \le i \le N-1} \mathbf{w}(i)$
$\mu = \min_{0 \le i \le N-1} \mathbf{w}(i)$
$\kappa = \frac{C}{\mu}$
$k = 0$
while $k < K_{\max}$ do
    $\nabla f(\mathbf{c}^k) = \mathbf{D}^H \mathrm{diag}(\mathbf{w}) \mathbf{D}(\mathbf{c}^k - \mathbf{x})$
    $\tilde{\mathbf{y}}^{k+1} = \mathbf{c}^k - \frac{1}{C} \nabla f(\mathbf{c}^k)$
    $\mathbf{y}^{k+1} = \Pi_Q(\tilde{\mathbf{y}}^{k+1})$
    Calculate $\gamma^{k+1}$ from $(\gamma^{k+1})^2 = (1 - \gamma^{k+1})(\gamma^k)^2 + \kappa \gamma^{k+1}$
    $\delta^k = \frac{\gamma^k(1-\gamma^k)}{(\gamma^k)^2 + \gamma^{k+1}}$
    $\mathbf{c}^{k+1} = \mathbf{y}^{k+1} + \delta^k(\mathbf{y}^{k+1} - \mathbf{y}^k)$
    $k = k + 1$
end while
$\mathbf{y}^* = \mathbf{y}^k$
```

Algorithm 7 Adapted optimal projected gradient algorithm

```
Input $\mathbf{x} \in \mathbb{R}^N$, $\mathbf{w} \in \mathbb{R}^N$, $K_{\max}$, $\boldsymbol{\delta} = [\delta^0 \delta^1 \dots \delta^{K_{\max}-1}]^T \in \mathbb{R}^{K_{\max}}$ $L$, $U$, $C^{-1}$
Output $\mathbf{y}^* \in \mathbb{R}^N$
$\mathbf{y}^0 = \mathbf{c}^0 = \mathbf{x}$
$\nabla f(\mathbf{c}^0) = \mathbf{0}$
$k = 0$
while $k < K_{\max}$ do
    $\tilde{\mathbf{y}}^{k+1} = \mathbf{c}^k - C^{-1} \nabla f(\mathbf{c}^k)$
    $\mathbf{y}^{k+1} = \Pi_Q(\tilde{\mathbf{y}}^{k+1})$
    $\mathbf{c}^{k+1} = \mathbf{y}^{k+1} + \delta^k(\mathbf{y}^{k+1} - \mathbf{y}^k)$
    $\nabla f(\mathbf{c}^{k+1}) = \mathbf{D}^H \mathrm{diag}(\mathbf{w}) \mathbf{D}(\mathbf{c}^{k+1} - \mathbf{x})$
    $k = k + 1$
end while
$\mathbf{y}^* = \mathbf{y}^k$
```

Figure 5.1 shows the block diagram of Algorithm 7 as it has been implemented onto FPGA, where each module has been coded using a specific C function. The main function `clip_top` consists of an iterative implementation of the per-iteration function `clip_iter`. In every iteration, the variables $\mathbf{c}^{k+1}$, $\mathbf{y}^{k+1}$ and $\nabla f(\mathbf{c}^{k+1})$ are stored in the respective memory blocks Memory 1, Memory 2 and Memory 3, in order to be accessed in the next iteration. The per-iteration function `clip_iter` consists of the following five subfunctions (the corresponding lines in Algorithm 2 are given in brackets):

- `clip_pipe` (lines 5-7) implements the gradient step, the projection onto the feasible set, and the computation of the weighted sum of previous iterates. It involves a sequence of element-by-element operations, that can be pipelined to reduce overall delay.
- `FFT` (line 8) implements the Fast Fourier Transform (FFT), for which the configuration parameters are set up in the function `config_fft`. Details on the FFT function an its configuration are given in subsection 5.3.2.
- `clip_weight` (line 8) implements the point-wise multiplication of the FFT output with the perceptual weights.
- `IFFT` (line 8) implements the inverse Fast Fourier Transform (IFFT), for which the configuration parameters are set up in the function `config_ifft`. Details on the IFFT function an its configuration are given in subsection 5.3.2.
- `clip_scale` performs a descaling of the IFFT output.

5.3 FPGA Implementation Aspects

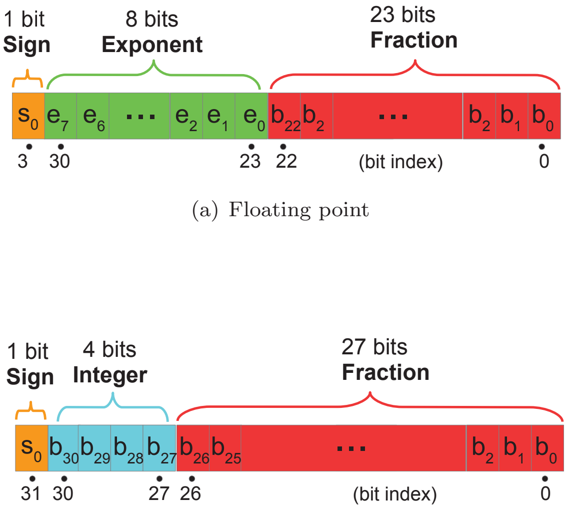
5.3.1 Floating-Point and Fixed-Point Arithmetic

FPGAs allow for a variety of computer arithmetic implementations, as they support both floating-point and fixed-point arithmetic. We will review here the main properties of floating-point and fixed-point arithmetic, and point out the pros and cons of using these arithmetics.

Floating-Point Arithmetic

The most commonly used floating point formats are described by the IEEE 754 standard [3]. As shown in Figure 5.2(a), the single-precision floating-point representation adopts a 32-bit architecture which uses 1 bit for the sign, 23 bits for the fractional part of the mantissa, and 8 bits for the signed exponent value, which moves the binary radix point with respect to the mantissa. The following relation holds between the single-precision floating-point value V_{float} and its corresponding bit representation:

$$V_{\text{float}} = \underbrace{(-1)^{s_0}}_{\text{sign}} \underbrace{M}_{\text{mantissa}} \underbrace{2^{E-127}}_{\text{exponential}} \tag{5.2}$$


1 bit
Sign
8 bits
Exponent
23 bits
Fraction
s0 e7 e6 ... e2 e1 e0 b22 b2 ... b2 b1 b0
3 30 23 22 (bit index) 0
1 bit
Sign
4 bits
Integer
27 bits
Fraction
s0 b30 b29 b28 b27 b26 b25 ... b2 b1 b0
31 30 27 26 (bit index) 0

(a) Floating point

(b) Fixed point

Figure 5.2: Floating and fixed point number representations.

where the mantissa M consists of a fixed integer part and a variable fractional part,

$$M = \underbrace{1}_{\text{integer}} + \underbrace{\sum_{i=1}^{23} b_{23-i} 2^{-i}}_{\text{fractional}} \tag{5.3}$$

and the decimal exponent E is

$$E = \sum_{i=0}^{7} e_i 2^i. \tag{5.4}$$

The most advantageous property of floating point arithmetic is that the exponent of each value can be selected to allow for maximum precision, and the exponent is stored with the fractional data. This allows to represent a wide range of numbers with a relatively small number of bits.

The main disadvantage is the need for additional hardware and longer delays for performing denormalization and normalization of the operands before and after every addition and subtraction [4] [5].

Fixed-Point Arithmetic

Fixed-point numbers use a fixed number of bits for the integer and the fractional parts. The exponent does not vary and does not have to be stored. The fixed-point example shown in Figure 5.2(b) adopts a 32-bit architecture, which uses 1 bit for the sign, 27 bits for the fractional part, and 4 bits for the integer part. In reality, an adder or any other arithmetic circuit does not know how many bits are interpreted as integer or fractional. This is purely an interpretational issue, so one can practically think of fixed-point arithmetic as integer arithmetic [6].

The following relation holds between the fixed-point value V_{fix} and its corresponding bit representation:

$$V_{\text{fix}} = \underbrace{\sum_{i=0}^{30} b_i 2^{i-27}}_{\text{integer+fractional}} \quad \underbrace{-s_0 2^4}_{\text{2's complement}} \tag{5.5}$$

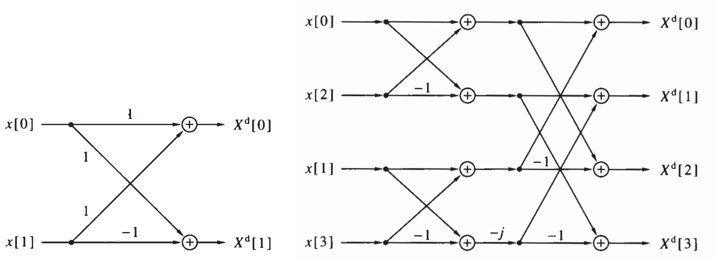
where the 2's complement format is generally used because it allows to perform subtraction and addition with the same hardware architecture.

The main advantage of using fixed-point implementations is the resulting lower cost and lower power, and often higher speed compared to floating-point implementations [2]. On FPGA platforms, there is the additional advantage of the fixed-point implementation being easily customizable to almost any desired word length [4].

The main disadvantage related to fixed-point implementations is the lower dynamic range and reduced precision compared to floating-point, which leads to different types of errors that can be incurred when performing fixed-point computations [1]:

- *Quantization errors*: these errors are introduced when high-precision input data is converted to the adopted reduced-precision fixed-point data format.
- *Overflow errors*: these errors are introduced whenever the number of bits for the integer part in the fixed-point representation is too small to represent a given number, e.g. after adding two numbers of like sign. In our proposed FPGA implementation, the number of integer bits was conservatively chosen large enough (5 bits) in order to avoid overflow.
- *Arithmetic errors*: these errors are introduced when multiplying two numbers having b fraction bits. The exact product can be represented using $2b$ fraction bits, so a b-bit truncation of the 2's complement number is necessary and introduces a round-off error.

Amongst the possible consequences of these errors occurring in the fixed-point implementation of an optimization method, are the loss of problem convexity,



(a) Radix-2 butterfly (b) Radix-4 butterfly

Figure 5.3: FFT butterfly structures [7].

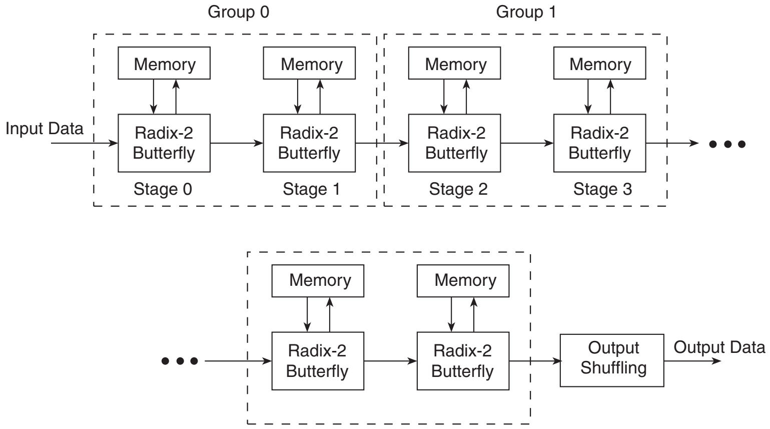


Figure 5.4: Pipelined Streaming I/O architecture [8].

the change of optimal solution, the lack of feasibility, and an unpredictable behaviour of the algorithm [1].

5.3.2 Fast Fourier Transform

For the implementation of the FFT we have used an IP core, i.e. a predeveloped block available from an FPGA vendor. We have used the Xilinx LogiCORE IP Fast Fourier Transform (FFT) v9.0, which implements the Cooley-Tukey FFT algorithm. This FFT core supports both fixed-point (8 to 34 bit) and single-precision floating-point (32 bit) arithmetic and is parametrized to allow for the run-time configuration of different FFT settings:

- The FFT mode, i.e. performing the forward or inverse complex FFT.

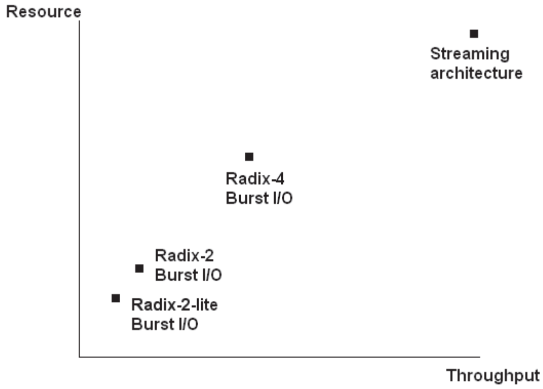


Figure 5.5: Resource usage versus throughput for FFT architecture options [8].

- The FFT transform size $N = 2^m$, $m = 3 - 16$.
- The scaling schedule, which represents the number of bits to be shifted after each stage of the FFT to avoid overflow, e.g. a scaling schedule of [2 3 2 0 1] bit-shifts (ordered from the first to the last stage) could be used for a five-stage implementation.

The FFT core provides four architecture options to trade-off resource usage and transform time. The basic blocks of all four FFT architectures are the so-called FFT Radix-r butterflies, which take in r complex data values and process them to produce a new set of r complex data values [9]. Figure 5.3(a) and Figure 5.3(b) show the structure of such a Radix-2 butterfly and a Radix-4 butterfly, respectively. The four resulting FFT architectures are the following:

- Pipelined Streaming I/O: from Figure 5.4, we observe that $\log_2 N$ Radix-2 butterfly processing stages are pipelined to offer continuous data processing. Each processing stage has its own memory banks to store input and intermediate data. The core has the ability to simultaneously perform transform calculations on the current frame of data, load input data for the next frame of data, and unload the results of the previous frame of data. The data is scaled after every pair of Radix-2 stages.
- Radix-4 Burst I/O: loads and processes data separately. The N-point FFT consists of $\log_4 N$ stages, with each stage containing $\frac{N}{4}$ Radix-4 butterflies. Transform sizes that are not a power of 4 need an extra Radix-2 stage for combining data.
- Radix-2 Burst I/O: loads and processes data separately. The N-point FFT consists of $\log_2 N$ stages, with each stage containing $\frac{N}{2}$ Radix-2 butterflies.
- Radix-2 Lite Burst I/O: a variant of Radix-2 Burst I/O with reduced

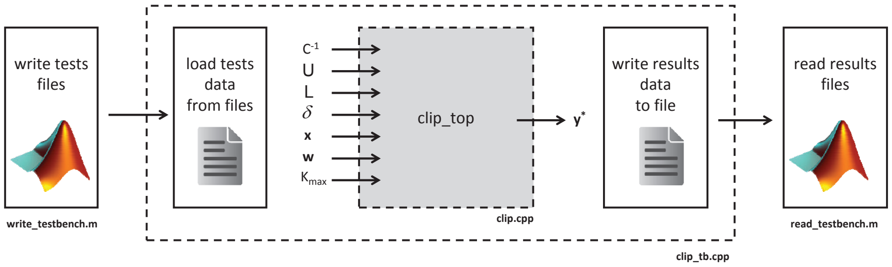


Figure 5.6: Data flow for the bit-accurate FPGA simulation in Vivado HLS.

resource usage at the expense of an additional delay.

The trade-off between resource usage and transform time for the different FFT architectures is illustrated in Figure 5.5. In order to minimize the transform time, we have selected to use the Pipelined Streaming I/O architecture, yet this choice comes with the cost of a higher resource usage. The following parameters have been selected:

- Transform size $N = 512 = 2^9$.
- Fixed conservative scaling schedule of [2 2 2 2 1] bit-shifts for the different FFT stages, which completely avoids overflows.

5.4 Simulation Results

5.4.1 Simulation Set-up

The C implementation of the architecture shown in Figure 5.1 has been transformed using Vivado HLS into a Register Transfer Level (RTL) implementation, which can be directly synthesized into any Xilinx FPGA.

In order to perform bit-accurate simulations of the RTL implementation, a simulation framework connecting Vivado HLS and MATLAB has been created, as shown in Figure 5.6. The input test data are stored into files using MATLAB (**write_testbench.m**), then read by the Vivado HLS bit-accurate simulation engine (**clip_tb.cpp**), and finally the simulation results are loaded into MATLAB (**read_testbench.m**). The simulation framework supports floating-point single-precision arithmetic and any fixed-point arithmetic up to 34 bits.

Two audio signals, `chopin.wav` and `mascagni.wav`, were selected to generate the input test data. The input test data per audio signal consists of all audio signal frames ($N = 512$) in which at least one sample exceeds a given amplitude range $[L, U]$, where the symmetric clipping level $U = -L$ was fixed at 0.14 and 0.25 for the respective audio signals. This procedure resulted in 86 and 233 audio signal frames constituting the input test data, respectively.

The RTL synthesis and bit-accurate simulations were performed for different settings of the applied arithmetic:

- Floating-point single-precision arithmetic (32 bits)
- Fixed-point arithmetic using $b = \{12,14,16,18,20,22,24,26,28,30,32,34\}$ fraction bits.

For every considered arithmetic setting, the bit-accurate simulations were repeated for different values of the fixed number of iterations $K_{\max}=\{10,20,30,40,50\}$ of the optimal projected gradient method.

5.4.2 Accuracy in Fixed-Point Arithmetic

We have computed the mean-squared error (MSE) between the output audio signal frame $\hat{\mathbf{y}}^*$ computed using bit-accurate hardware simulations in fixed-point or single-precision floating point arithmetic, and the reference output audio signal frame $\mathbf{y}^*$ computed using double-precision floating-point arithmetic in MATLAB,

$$\mathrm{MSE}(\hat{\mathbf{y}}^*) = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i^* - y_i^*)^2 \tag{5.6}$$

In Figures 5.7(a) and 5.7(b), the MSE for all audio signal frames is plotted for the different fixed-point and floating-point arithmetics, for $K_{\max} = 50$. For both audio signals, we observe a fairly constant MSE over all audio signal frames for a given arithmetic choice. The per-frame MSE in fixed-point arithmetic is seen to improve by roughly a decade per two additional fraction bits. When using the maximum of 34 fixed-point fraction bits, the MSE values approach the MSE in single-precision floating point, which is as low as 10^{-14}.

The MSE gives an indication of the per-frame accuracy, which is not sufficient to draw conclusions on the quality of the complete audio signal. In order to assess which is the minimum number of fixed-point fraction bits to be used from an audio quality point of view, the following analysis has been conducted.

The objective audio quality difference for each audio signal is assessed by computing the ΔODG measure,

$$\Delta\mathrm{ODG} = \mathrm{ODG}(\mathbf{x_s}, \mathbf{y_s^*}) - \mathrm{ODG}(\mathbf{x_s}, \hat{\mathbf{y}}_\mathbf{s}^*) \tag{5.7}$$

where $\mathbf{x_s}$ is the input audio signal, $\mathbf{y_s^*}$ is the reference output audio signal computed using double-precision floating-point arithmetic in Matlab, and $\hat{\mathbf{y}}_\mathbf{s}^*$

$K_{\max}$ — b	12	16	20	24	28	32	34
10	0.803	0.245	0.070	0.045	0.046	0.046	0.046
20	0.804	0.549	0.036	0.025	0.023	0.023	0.023
30	0.797	0.417	**0.011**	**0.011**	**0.009**	**0.009**	**0.009**
40	0.793	0.466	0.022	**0.001**	**0.003**	**0.003**	**0.002**
50	0.786	0.497	0.042	**0.002**	**0.000**	**0.000**	**0.000**

Table 5.1: ΔODG audio quality difference with respect to double precision implementation, with b the number of fixed-point fraction bits employed and $K_{\max}$ the fixed number of iterations, for fragment `chopin.wav`. Imperceptible differences in bold.

$K_{\max}$ — b	12	16	20	24	28	32	34
10	0.254	0.120	0.058	0.049	0.049	0.049	0.049
20	0.254	0.057	0.027	0.021	0.021	0.021	0.021
30	0.255	0.099	**0.006**	**0.006**	**0.006**	**0.006**	**0.006**
40	0.254	0.121	**0.004**	**0.001**	**0.001**	**0.000**	**0.000**
50	0.254	0.135	**0.012**	**0.001**	**0.000**	**0.000**	**0.000**

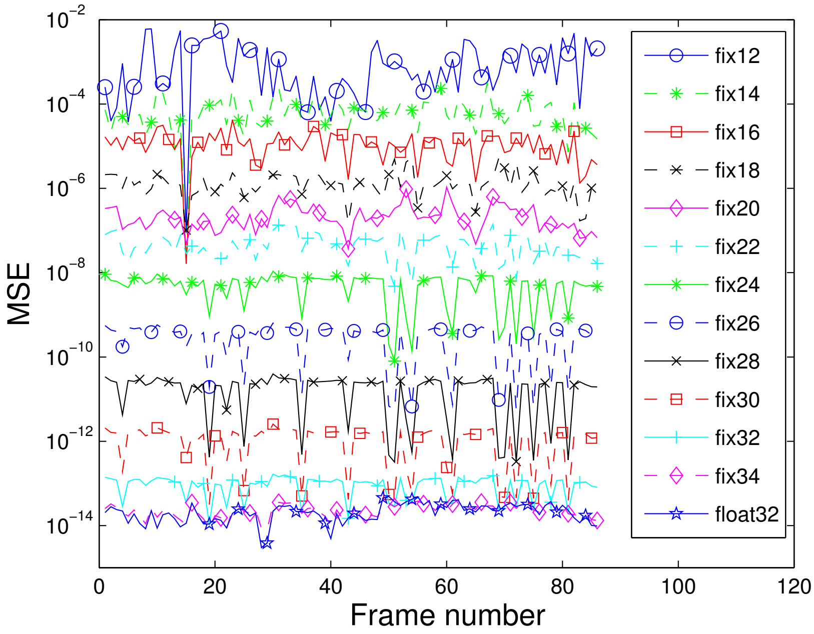
Table 5.2: ΔODG audio quality difference with respect to double precision implementation, with b the number of fixed-point fraction bits employed and $K_{\max}$ the fixed number of iterations, for fragment `mascagni.wav`. Imperceptible differences in bold.

is the output audio signal computed using bit-accurate hardware simulations in fixed-point or single-precision floating point arithmetic. ODG($\mathbf{r}, \mathbf{d}$) is an objective measure [10] which predicts the audio quality of a signal $\mathbf{d}$ with respect to a signal $\mathbf{r}$ on a scale of $[0, -4]$, where 0 corresponds to an imperceptible degradation, and -4 corresponds to a very annoying degradation.

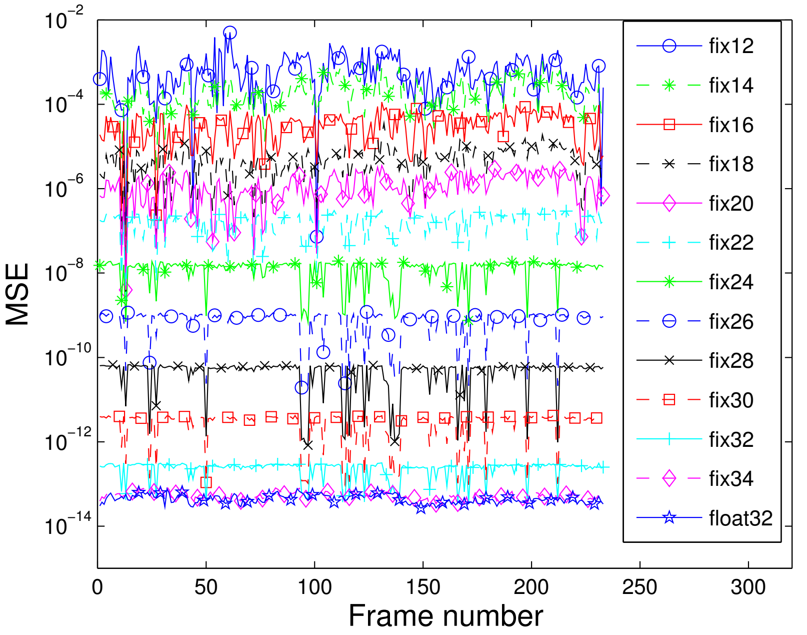
In Tables 5.1 and 5.2, the ΔODG audio quality difference is given for different values of the number of fixed-point fraction bits b and the number of iterations $K_{\max}$. An interesting observation is that in some cases, the audio quality difference increases with the number of iterations due to the increasing accumulation of round-off errors. As the resolution of the ODG is 0.02 points [10], we can interpret differences falling within 0.02 ODG points as being imperceptible (boldfaced values). For both audio signals, it is seen to be sufficient to perform $K_{\max} = 30$ projected gradient iterations using $b = 20$ fixed-point fraction bits in order to avoid perceptible artefacts.

5.4.3 Latency, Resource Usage and Power Consumption

Depending on the employed arithmetic, the FPGA design will exhibit different characteristics in terms of overall latency, resource usage and power consumption. We have assessed these relations using Vivado HLS, selecting a Xilinx



(a) `chopin.wav`



(b) `mascagni.wav`

Figure 5.7: MSE per audio signal frame for different fixed-point and floating-point arithmetics, $K_{\max} = 50$.

Artix-7 FPGA as the target device and clocked at a frequency of 200MHz.

Latency

The overall latency c_{total} of the design is the number of clock cycles from the start of execution until the final output is written. For our algorithm, the overall latency c_{total} can be computed as a function of the number of clock cycles per iteration c_{iter},

$$c_{\text{total}} = 2N + K_{\max}(N + c_{\text{iter}}) \tag{5.8}$$

which accounts for overhead clock cycles spent on reading in and writing back results.

For our design using $b = 20$ fixed-point fraction bits and $K_{\max} = 30$ iterations, the observed number of clock cyles per iteration $c_{\text{iter}} = 4900$ and the overall latency $c_{\text{total}} = 163384$. At a clocking frequency of 200MHz, this corresponds to a per-iteration time latency of 25μs, and an overall time latency of 817μs.

Resource Usage

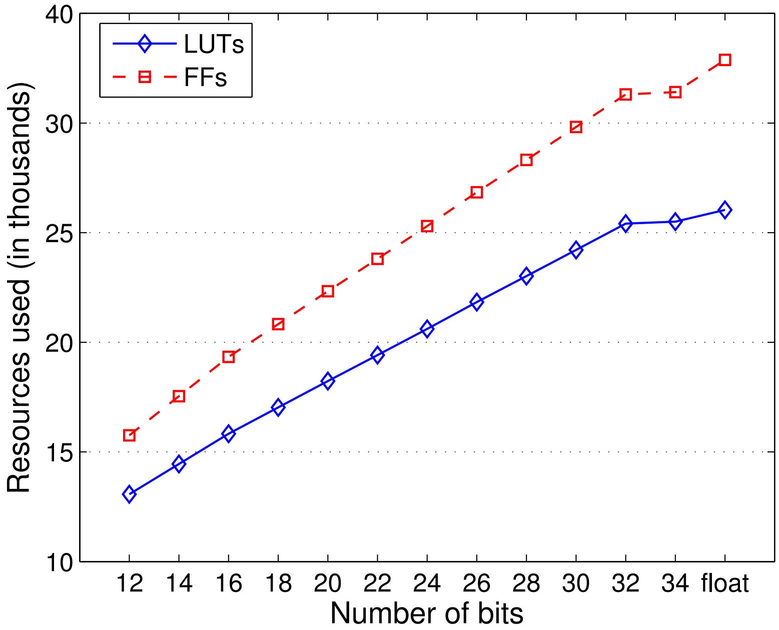
Figure 5.8(a) shows the number of look-up tables (LUTs) and Flip-Flops (FFs) used to implement the design for different fixed-point and floating-point arithmetics. In fixed-point arithmetic, we observe a linearly increasing FF/LUT usage for increasing number of fixed-point fraction bits. The design for the single-precision floating point arithmetic uses slightly more resources compared to the fixed-point desing with the maximum 34 fraction bits.

Figure 5.8(b) shows the number of BRAM memory blocks and DSP48E slices used to implement the design for different fixed-point and floating-point arithmetics. In fixed-point arithmetic, the relation between the number of fixed-point fraction bits and the BRAM/DSP48E usage resembles a staircase function, of which the first stair is seen to occur around 18 bits, and the second stair around 27 bits. This can be explained by the fact that FPGA multipliers are 18 bits wide, and on-chip memory blocks are multiples of 9 bits wide.

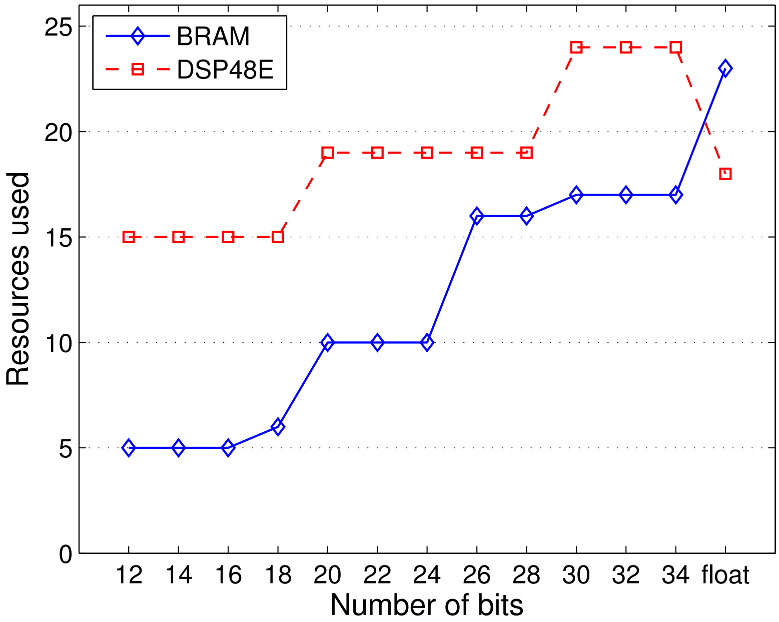
Compared to single-precision floating point arithmetic design, the use of a fixed-point arithmetic design using $b = 20$ fixed-point fraction bits is seen to result in a reduction of FF usage (-32%), LUT usage (-29%), DSP48E usage (-56%) for a similar BRAM usage.

Power Consumption

The power consumption of an FPGA can be a critical design constraint, especially for mobile applications. In order to estimate the power dissipation of the target FPGA device, two main power dissipation sources must be considered:



(a) FFs and LUTs



(b) DSP48E and BRAM

Figure 5.8: Resource usage for different fixed-point and floating-point arithmetics.

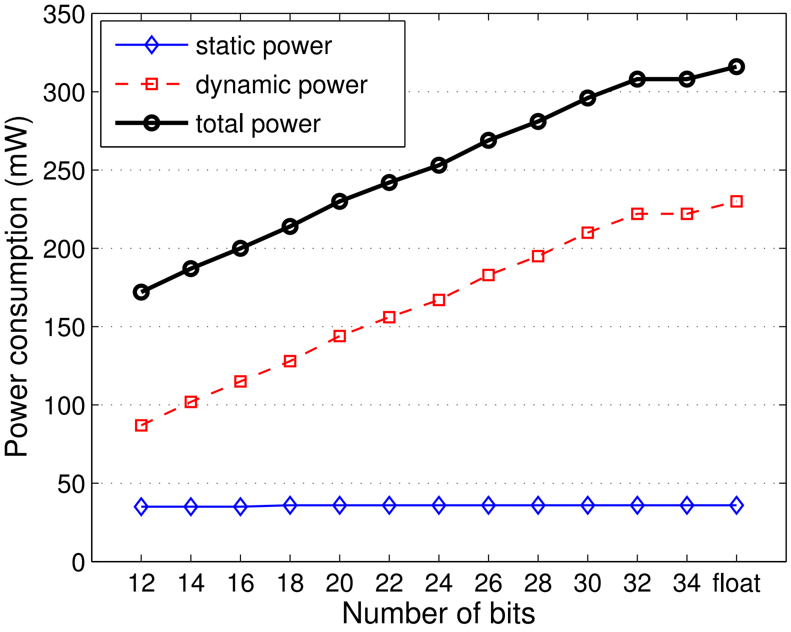


Figure 5.9: Static, dynamic and total power consumption estimates for different fixed-point and floating-point arithmetics on a Xilinx Artix-7 FPGA.

1. Static power dissipation: this is device-dependent and design-independent.
2. Dynamic power dissipation: this depends mainly on the clock frequency and on the number of logic cells and other resources in use, and on the average percentage of cells toggling at each clock cycle.

An estimate of both the static and dynamic power consumption was computed using the Xilinx Power Estimator (XPE) tool. In Figure 5.9, the static, dynamic and total power consumption estimates are shown for different fixed-point and floating-point arithmetics on a Xilinx Artix-7 FPGA. In fixed-point arithmetic, we observe a linearly increasing dynamic power consumption for increasing number of fixed-point fraction bits, and a constant static power consumption.

The use of a fixed-point arithmetic design using $b = 20$ fixed-point fraction bits is seen to result in a total power consumption of 280 mW, which constitutes a reduction of 23 % compared to a single-precision floating point arithmetic design.

Available on `http://www.xilinx.com/products/design_tools/logic_design/xpe.htm`.

5.5 Conclusions

This chapter has considered an embedded hardware implementation on an FPGA of the optimal projected gradient method, which forms the core of the perception-based clipping algorithm. It was demonstrated that the choice between fixed-point arithmetic and floating-point arithmetic and the selection of the corresponding bit width has an impact on the resulting audio quality as well as on the resource usage and power consumption of the design.

The resulting FPGA design, which performs 30 projected gradient iterations using 20 fixed-point fraction bits, has a low latency, a reduced power consumption, a reduced resource usage, and at the same time preserves the full audio quality improvement. This design thus allows to perform the algorithm in real time on a small and portable device.

Bibliography

[1] J. L. Jerez, P. J. Goulart, S. Richter, G. A. Constantinides, E. C. Kerrigan, and M. Morari, "Embedded online optimization for model predictive control at megahertz rates," *Submitted to IEEE Transactions on Automatic Control*, 2013.

[2] U. Meyer-Baese, *Digital Signal Processing with Field Programmable Gate Arrays.* Springer, 2007.

[3] IEEE, "IEEE standard for binary floating-point arithmetic," *Std 754-1985*, 1985.

[4] W. T. Padgett and D. V. Anderson, "Fixed-point signal processing," *Synthesis Lectures on Signal Processing*, vol. 4, no. 1, pp. 1–133, 2009.

[5] S. W. Smith *et al.*, *The scientist and engineer's guide to digital signal processing.* California Technical Pub. San Diego, 1997.

[6] J. Serrano, "Introduction to FPGA design," in *8th Workshop on Electronics for LHC Experiments.* CERN, 2004, pp. 231–247.

[7] B. Porat, *A course in digital signal processing.* Wiley New York, 1997.

[8] "LogiCORE IP Fast Fourier Transform v9.0 Product Guide for Vivado Design Suite," http://www.xilinx.com/support/documentation/ip_documentation/xfft/v9_0/pg109-xfft.pdf, accessed: 2013-10-21.

[9] P. Kabal and B. Sayar, "Performance of fixed-point FFT's: Rounding and scaling considerations," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 11, 1986, pp. 221–224.

[10] International Telecommunications Union Recommendation BS.1387, "Method for objective measurements of perceived audio quality," 1998.

Part III

Recovery Algorithms

Chapter 6

Declipping Using Perceptual Compressed Sensing

Declipping of Audio Signals Using Perceptual Compressed Sensing

Bruno Defraene, Naim Mansour, Steven De Hertogh,
Toon van Waterschoot, Moritz Diehl, and Marc Moonen

Published in *IEEE Trans. Audio Speech Language Process.*, vol. 21, no. 12, Dec. 2013, pp. 2627-2637.

Contributions of first author

- literature study
- co-design of perceptual compressed sensing framework
- co-development of PCSL1 declipping algorithm
- co-design of objective evaluation experiments
- software implementation and computer simulations
- co-interpretation of objective evaluation results
- co-design and supervision of subjective listening tests
- co-analysis and co-interpretation of subjective evaluation results
- co-formulation of conclusions
- text redaction and editing

Abstract

The restoration of clipped audio signals, commonly known as *declipping*, is important to achieve an improved level of audio quality in many audio applications. In this paper, a novel declipping algorithm is presented, jointly based on the theory of compressed sensing (CS) and on well-established properties of human auditory perception. Declipping is formulated as a sparse signal recovery problem using the CS framework. By additionally exploiting knowledge of human auditory perception, a novel perceptual compressed sensing (PCS) framework is devised. A PCS-based declipping algorithm is proposed which uses ℓ_1-norm type reconstruction. Comparative objective and subjective evaluation experiments reveal a significant audio quality increase for the proposed PCS-based declipping algorithm compared to CS-based declipping algorithms.

6.1 Introduction

Clipping introduces undesired signal distortion in many audio applications. Clipping can occur both in the analog domain and in the digital domain, and is generally caused by the inability of an audio playback, recording or processing device to deliver the dynamic range required by the audio signal. When a digital audio signal is clipped, all its sample values lying beyond a maximum amplitude level (referred to as the *clipping level* θ_c) are mapped onto $\pm\theta_c$, as shown in Figure 6.1.

Clipping inevitably introduces nonlinear distortion into the audio signal, consisting of both the *modification* of existing frequency components, and the *introduction* of new harmonic and aliasing frequency components into the signal [1]. In a series of listening experiments [2], it has been concluded that clipping has a significant negative effect on the perceived audio quality. More specifically, clipping is typically associated with the perceptible addition of crackling noises to the original audio signal, often qualified as (very) annoying.

Therefore, performing *declipping*, i.e. the restoration of the clipped audio signal, is necessary to achieve an improved level of audio quality and listener satisfaction. In past research contributions, several approaches to the declipping problem have been proposed. A first approach is based on performing an interpolation/extrapolation step to recover the clipped signal portions based on the knowledge of unclipped signal portions [3]. A second approach consists in adopting a suitable model of both the clean signal (typically autoregressive) and the clipping distortion, and subsequently recovering the clean signal

In the context of this research, digital *hard* clipping is considered, and it will be simply termed "clipping" throughout this paper. Note that in analogue systems, *soft* clipping is very common.

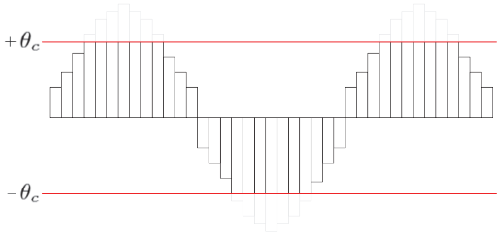


Figure 6.1: Clipping of a digital audio signal.

through Bayesian statistical estimation of its model parameters [4]. Other notable declipping approaches include the use of artificial neural networks [5].

Recently, the declipping problem has been adressed in the generic framework of compressed sensing (CS), and encouraging results have been reported [6, 7, 8, 9]. In the CS framework, declipping is formulated and solved as a sparse signal recovery problem, where one takes advantage of the sparsity of the clean audio signal (in some basis) in order to recover it from a subset of its samples.

In this paper, we propose a perceptual compressed sensing (PCS) framework, in which a declipping algorithm is subsequently developed. The PCS framework integrates CS theory and properties of human auditory perception. By incorporating knowledge of human auditory perception in the algorithm design, it is possible to further improve the perceptual reconstruction quality. The advantage of using perceptual knowledge has been recently demonstrated in related audio applications, such as the sparse approximation of audio signals [10], and the limiting of audio signals [11].

The paper is organized as follows. In Section 6.2, the basic principles of CS as well as its fundamental limitations are reviewed, and the declipping problem is subsequently formulated in the CS framework. In Section 6.3, the PCS framework is presented, leading to a PCS-based declipping algorithm, using ℓ_1-norm type reconstruction. In Section 6.4, a comparative objective and subjective evaluation of different declipping algorithms is discussed. Finally, in Section 6.5, some concluding remarks are presented.

6.2 A CS Framework for Declipping

The theory of compressed sensing has been introduced in [12] [13] [14] in the context of sparse signal sampling and acquisition. Basically, the theory states that any signal that can be considered sparse in some basis, can be uniquely, and in many cases *perfectly* reconstructed based on sub-Nyquist rate sampled measurements. This notion goes against the commonly accepted Nyquist sampling criterion. In the next subsections, we review the basic principles of CS as well as its fundamental limitations, and subsequently formulate the declipping problem in the CS framework.

6.2.1 CS Basic Principles

To be able to outline the basic principles of CS, we first introduce some necessary definitions and notation. Signals are considered to be real-valued vectors in an N-dimensional normed Euclidean vector space $\mathbb{R}^N$. For the purposes of CS, mainly the ℓ_0 and ℓ_1 norm are of importance. The *sparsity* k of a vector $\mathbf{x}$ is defined as the number of non-zero components, i.e. $\|\mathbf{x}\|_0 = k$. A vector $\mathbf{x}$ with sparsity k is said to be k*-sparse*.

Compressed sensing takes advantage of the sparsity of a signal $\mathbf{s} \in \mathbb{R}^N$ in some fixed basis $\mathbf{\Psi} \in \mathbb{C}^{N\times N}$ in order to recover it from a reduced measurement $\mathbf{y} \in \mathbb{R}^M$, where $M < N$. Acquiring a signal $\mathbf{s}$ by CS consists of two main steps:

1. *Measurement*: apply a measurement matrix $\mathbf{\Phi}$ to obtain the measurement $\mathbf{y}$,

$$\mathbf{y} = \mathbf{\Phi}\mathbf{s} = \mathbf{\Phi}\mathbf{\Psi}\mathbf{x} = \mathbf{A}\mathbf{x} \tag{6.1}$$

   with $\mathbf{s} \in \mathbb{R}^N$ the original signal, $\mathbf{x} \in \mathbb{R}^N$ its sparse decomposition, $\mathbf{\Psi} \in \mathbb{C}^{N\times N}$ the fixed basis, $\mathbf{\Phi} \in \mathbb{R}^{M\times N}$ the measurement matrix, $\mathbf{A} = \mathbf{\Phi}\mathbf{\Psi} \in \mathbb{C}^{M\times N}$ the sensing matrix, and $\mathbf{y} \in \mathbb{R}^M$ the measurement.

2. *Recovery*: (in)exactly recover $\mathbf{s}'$ from $\mathbf{y}$ using constrained ℓ_p-norm optimization,

$$\mathbf{x}' = \arg\min_{\mathbf{z}} \|\mathbf{z}\|_p \quad \text{subject to} \quad \mathbf{A}\mathbf{z} = \mathbf{y} \tag{6.2}$$

$$\mathbf{s}' = \mathbf{\Psi}\mathbf{x}' \tag{6.3}$$

   with $p \in \{0, 1\}$ in typical CS applications.

The actual choice of the ℓ_p norm in the recovery step (6.2), has considerable implications on the resulting solution $\mathbf{x}'$ as well as on its computation. Firstly, using an ℓ_0 or ℓ_1 norm in the minimization problem will obviously lead to a sparser solution $\mathbf{x}'$ compared to using ℓ_p norms with $p > 1$. This is expected to provide a more accurate approximation for the sparse signal under consideration. Secondly, optimization problem (6.2) has distinct properties depending on the ℓ_p norm used in the minimization, and consequently requires distinct

In this paper, we restrict the analysis to fixed bases $\mathbf{\Psi} \in \mathbb{C}^{N\times N}$, and do not consider the case of overcomplete dictionaries $\mathbf{\Psi} \in \mathbb{C}^{N\times K_D}$ with $N < K_D$. Overcomplete dictionaries have been used to allow for sparser representations of audio signals [15].

optimization methods. Considering ℓ_0-norm minimization, the resulting optimization problem is non-convex, implying that one has to rely on greedy methods such as orthogonal matching pursuit (OMP) [16]. On the other hand, ℓ_1-norm minimization also induces sparsity in the solution (albeit to a lesser extent than ℓ_0-norm minimization) and has the advantage of leading to a convex optimization problem, which can be solved by convex optimization methods or dedicated algorithms such as Basis Pursuit (BP) [17] or Dantzig selector [18]. It is also possible to consider ℓ_p norms with $0 < p < 1$, again leading to non-convex optimization problems.

6.2.2 Perfect Recovery Guarantees

We now review the sufficient conditions under which it is possible to perfectly recover a k-sparse signal through CS as introduced in (6.1)-(6.3). The sensing matrix $\mathbf{A}$ will be of crucial importance, as the sufficient conditions for perfect recovery will be based on its properties. We remark that the sufficient conditions in compressed sensing are generally not satisfied in practical applications, and that their main interest is theory rather than applicability.

Spark Property

A first important requirement is that the sensing matrix $\mathbf{A}$ maps all distinct k-sparse vectors $\mathbf{x_1} \neq \mathbf{x_2}$ onto distinct vectors $\mathbf{y_1} \neq \mathbf{y_2}$. For an exact k-sparse $\mathbf{x}$, this holds if and only if

$$\text{spark}(\mathbf{A}) > 2k \tag{6.4}$$

where spark($\mathbf{A}$) is defined as the smallest number of columns of $\mathbf{A}$ that are linearly dependent.

Restricted Isometry Property (RIP) [19]

This property provides a more subtle recovery guarantee in the case of noisy measurements. Essentially, a matrix $\mathbf{A}$ satisfies the RIP of order k if there exists a $\delta_k \in (0, 1)$ such that

$$(1 - \delta_k)\|\mathbf{x}\|_2^2 \leq \|\mathbf{Ax}\|_2^2 \leq (1 + \delta_k)\|\mathbf{x}\|_2^2 \tag{6.5}$$

holds for all k-sparse $\mathbf{x}$. If $\mathbf{A}$ satisfies the RIP of order $2k$ with $0 < \delta_{2k} \leq 0.5$, this is a sufficient condition for a variety of CS algorithms to be able to perfectly recover any k-sparse signal.

Coherence Property

In practice, the spark and RIP properties are difficult to compute for a given sensing matrix $\mathbf{A}$. The coherence property provides a more practically feasible way to establish recovery guarantees. The *coherence* μ of a matrix $\mathbf{A}$ is defined as [20],

$$\mu(\mathbf{A}) = \max_{i,j,i\neq j} \frac{|\mathbf{a}_i^H \mathbf{a}_j|}{\|\mathbf{a}_i\|_2 \|\mathbf{a}_j\|_2} \tag{6.6}$$

where $\mathbf{a}_i$ denotes the i-th column of $\mathbf{A}$. The coherence of a matrix is a measure of the decorrelation it provides as a transformation between the original and the analysis domain. The coherence property is related to the spark property through [21],

$$\text{spark}(\mathbf{A}) \geq 1 + \frac{1}{\mu(\mathbf{A})} \tag{6.7}$$

Combining (6.7) and (6.4), and assuming no noise on the measurements, this leads to the sufficent condition

$$k < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{A})}\right) \tag{6.8}$$

for perfect recovery of a k-sparse signal using ℓ_0 and ℓ_1- norm minimization to be possible.

6.2.3 CS-Based Declipping

The CS framework is a suitable framework for addressing the declipping problem. Because of its ability to reconstruct a sparse signal from a reduced measurement, CS can theoretically recover the original signal, including the parts that were clipped and deemed “lost”. CS-based declipping is based on the following principles:

Measurement Matrix $\mathbf{\Phi}$

The measurement $\mathbf{y} = \mathbf{\Phi s}$ simply consists of the M unclipped samples in the signal $\mathbf{s}$. The measurement matrix $\mathbf{\Phi}$ is entirely defined by the clipping pattern, i.e. it is a submatrix of an identity matrix, where the rows corresponding to the clipped samples have been removed.

Fixed Basis $\mathbf{\Psi}$

Appropriate sparsifying time-domain signal decompositions for audio signals include the Discrete Fourier Transform (DFT) and the Discrete Cosine Transform (DCT), which are defined as

$$X_l^{\text{DFT}} = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x_n e^{-j2\pi \frac{l}{N} n}, \quad l = 0 \ldots N-1 \tag{6.9}$$

$$X_l^{\mathrm{DCT}} = \alpha(l,N)\sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)l\right] \tag{6.10}$$

$$\text{with} \quad \alpha(l,N) = \frac{\sqrt{1+\min(l,1)}}{\sqrt{N}}\ ,\ l = 0\dots N-1 \tag{6.11}$$

where x_n denotes the n-th element of the time-domain vector $\mathbf{x} \in \mathbb{R}^N$, and X_l denotes the l-th element of the transform domain vector $\mathbf{X} \in \mathbb{R}^N$. The DFT and DCT bases are suitable candidates to serve as the fixed basis in the declipping problem (i.e., $\boldsymbol{\Psi}$ can be chosen as the Inverse DFT matrix or the inverse DCT matrix). However, the DCT has an advantage compared to the DFT in that it involves only real-valued calculations.

Perfect Recovery Guarantees

The RIP is a very restrictive condition, which can only hold for specific classes of sensing matrices. One such class is the class of *random partial orthogonal matrices*, i.e matrices obtained by randomly choosing M rows from a normalized orthogonal matrix [22]. For this class of matrices, the following result applies.

Theorem 6.1 *[23] Given an orthogonal matrix $\boldsymbol{\Psi}$ with entries of magnitude $O(\frac{1}{\sqrt{N}})$. A sensing matrix $\mathbf{A} = \boldsymbol{\Phi}\boldsymbol{\Psi}$ consisting of a random subset of M rows of $\boldsymbol{\Psi}$ satisfies the RIP condition $\delta_{3k} + 3\delta_{4k} \leq 2$ with high probability if and only if*

$$M \geq O(k \log N). \tag{6.12}$$

For the declipping problem under consideration, this is presumably the most relevant perfect recovery guarantee that can be given. However, this perfect recovery guarantee only holds under the assumption of random positions of the clipped samples, which is not expected to be met for most clipped audio signals encountered in real-world audio devices. Moreover, even if this assumption is met, it is very likely that the RIP condition does not hold for dictionaries commonly used and values of sparsity k commonly encountered in audio signal declipping.

Frame-Based Processing

Because of the short-time stationarity of audio signals, a declipping algorithm should operate on short-time signal frames. In the declipping algorithms proposed in this paper, a clipped audio signal is first split into Hann-windowed half-overlapping frames of N samples each. The frames are sequentially declipped, and synthesized to form the declipped audio signal. Introducing the subscript m as the frame index, the recovery step (6.2) for the m-th frame can be rewritten as

$$\mathbf{x}'_m = \arg\min_{\mathbf{z}} \|\mathbf{z}\|_p \text{ s.t. } \mathbf{A}_m\mathbf{z} = \mathbf{\Phi}_m\mathbf{\Psi}\mathbf{z} = \mathbf{y}_m. \tag{6.13}$$

Alternative Recovery Step

The structure of a clipped audio signal has an inherent advantage, pertaining to the CS recovery step. As mentioned before, when applying a clipping level θ_c to a signal, all sample values beyond θ_c are mapped onto θ_c. This means that the original value of any clipped sample is in absolute value *larger than or equal to* θ_c. This forms an additional constraint on the eligible solution space for the sparse recovery [7], which leads to the following recovery step,

$$\mathbf{x}'_m = \arg\min_{\mathbf{z}} \|\mathbf{z}\|_p \text{ s.t. } \begin{cases} \mathbf{A}_m\mathbf{z} = \mathbf{\Phi}_m\mathbf{\Psi}\mathbf{z} = \mathbf{y}_m \\ \mathbf{C}_m^{+}\mathbf{\Psi}\mathbf{z} \geq \theta_c \\ \mathbf{C}_m^{-}\mathbf{\Psi}\mathbf{z} \leq -\theta_c \end{cases} \tag{6.14}$$

where $\mathbf{C}_m^{+}$ and $\mathbf{C}_m^{-}$ are submatrices of an identity matrix where the rows corresponding to positively and negatively clipped samples, are respectively selected. Adding these constraints to the optimization problem is expected to improve the signal recovery quality.

Relaxation of Equality Constraints

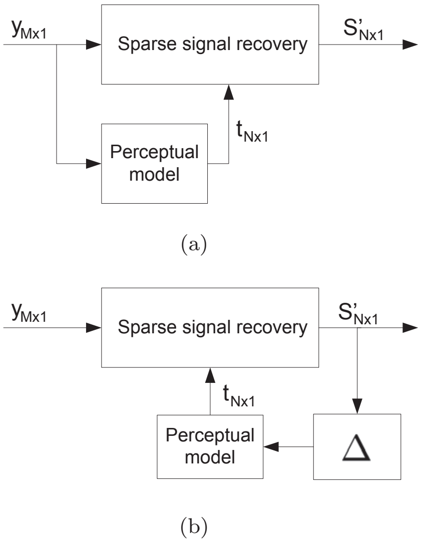
When the signal $\mathbf{s}$ is not exactly sparse but only compressible, as it is the case for most real-world audio signals, a certain relaxation to the equality constraints $\mathbf{A}_m\mathbf{z} = \mathbf{y}_m$ is desirable [24]. For ℓ_0-norm minimization, a possible relaxation is to solve the following optimization problem,

$$\mathbf{x}'_m = \arg\min_{\mathbf{z}} \|\mathbf{z}\|_0 \text{ s.t. } \begin{cases} \|\mathbf{y}_m - \mathbf{A}_m\mathbf{z}\|_2^2 \leq \epsilon_m \\ \mathbf{C}_m^{+}\mathbf{\Psi}\mathbf{z} \geq \theta_c \\ \mathbf{C}_m^{-}\mathbf{\Psi}\mathbf{z} \leq -\theta_c \end{cases} \tag{6.15}$$

where the value of the parameter $\epsilon_m > 0$ should be carefully selected [7]. As opposed to the ℓ_0-norm minimization based declipping algorithms presented in [7], we will focus on the use of ℓ_1-norm minimization for the declipping of audio signals. When using ℓ_1-norm minimization, the relaxation proposed here is to solve the following optimization problem,

$$\mathbf{x}'_m = \arg\min_{\mathbf{z}} \|\mathbf{z}\|_1 + \frac{1}{2\gamma_m}\|\mathbf{y}_m - \mathbf{A}_m\mathbf{z}\|_2^2 \text{ s.t. } \begin{cases} \mathbf{C}_m^{+}\mathbf{\Psi}\mathbf{z} \geq \theta_c \\ \mathbf{C}_m^{-}\mathbf{\Psi}\mathbf{z} \leq -\theta_c \end{cases} \tag{6.16}$$

where $\gamma_m > 0$ is a regularisation parameter. It is interesting to note that this optimization problem is similar to the well-known Basis Pursuit Denoising problem [17], except for the additional inequality constraints.



(a)

(b)

Figure 6.2: Schematic overview of perceptual compressed sensing: (a) Feedforward mode (b) Feedback mode. The symbol Δ denotes a one-frame delay.

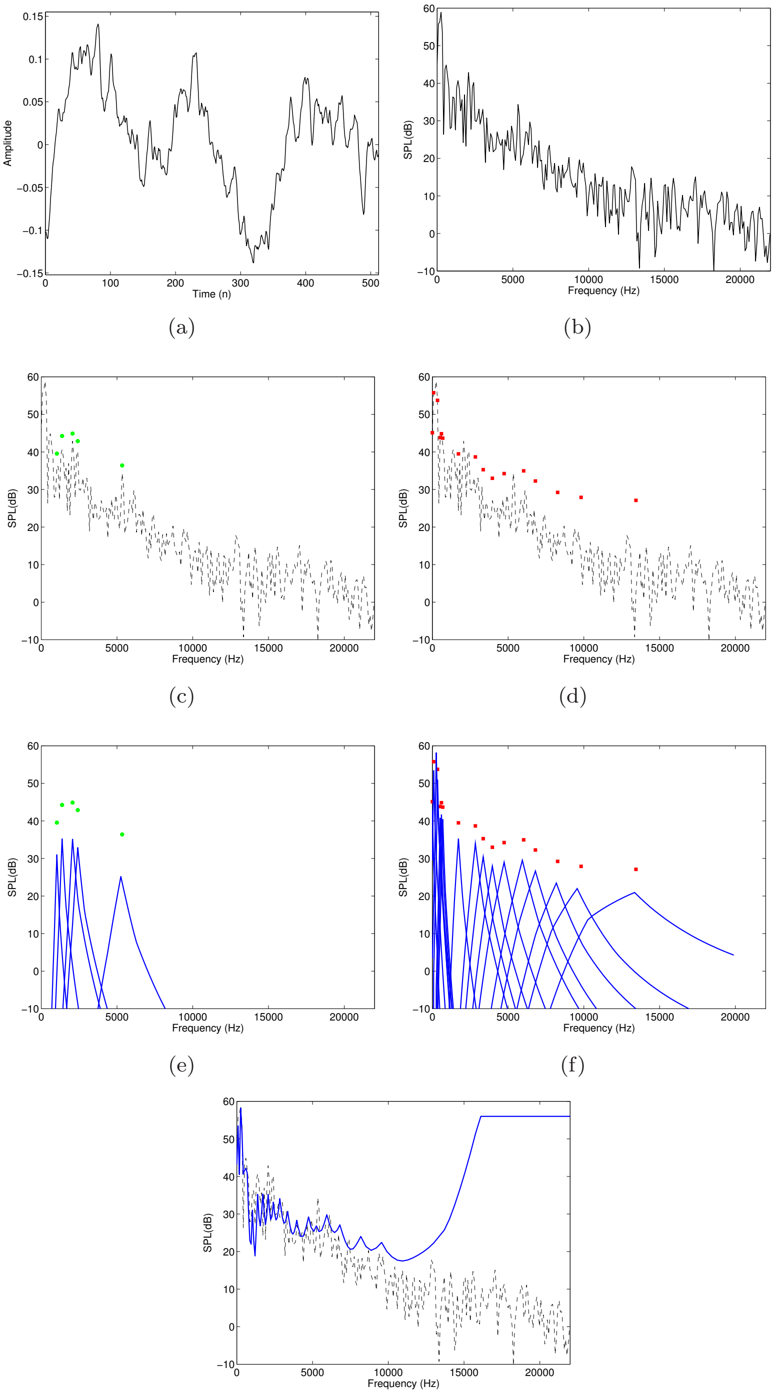
6.3 A PCS Framework for Declipping

In Section 6.2, the CS framework has been shown to be a suitable framework for addressing the declipping problem. However, of crucial importance for audio applications is the resulting *perceived audio quality* of the declipped signal, which does not necessarily coincide with the physical signal reconstruction quality. By additionally incorporating knowledge of human auditory perception, a novel perceptual compressed sensing (PCS) framework is presented in Subsections 6.3.1 and 6.3.2. A PCS-based declipping algorithm, using ℓ_1-norm type reconstruction, is presented in Subsection 6.3.3.

6.3.1 Perceptual CS Framework

It is known that audio signal components at certain frequencies are more perceptible than components at other frequencies, and that the relative perceptibility is partly signal-dependent. Two phenomena of human auditory perception are responsible for this:

- The *absolute threshold of hearing* is defined as the required intensity (dB) of a pure tone such that an average listener will just hear the tone in a noiseless environment. The absolute threshold of hearing is a function of



(a) (b)

(c) (d)

(e) (f)

(g)

Figure 6.3: Different steps in the computation of the global masking threshold using the ISO/IEC 11172-3 MPEG-1 Layer 1 psychoacoustic model 1 : (a)-(b) Time domain and normalized frequency domain representations of the input audio signal (c)-(d) Tonal maskers (circles), non-tonal maskers (squares) and input frequency spectrum (dotted line) (e)-(f) Individual masking thresholds related to tonal and non-tonal maskers respectively (g) Global masking threshold (solid line) and input frequency spectrum (dotted line) [11].

the tone frequency and has been measured experimentally [25].

- *Simultaneous masking* is a phenomenon where the presence of certain spectral energy (the masker) masks the simultaneous presence of weaker spectral energy (the maskee), or in other words, renders it imperceptible.

Combining both phenomena, the *instantaneous global masking threshold* $\mathbf{t}$ of an audio signal gives the amount of signal energy (dB) at each frequency that can be masked by the rest of the signal. As such, the masking threshold $\mathbf{t}$ gives an indication of the relative perceptibility of signal components at different frequencies, and can be used in the CS framework in order to focus on the recovery of perceptually important signal components, while at the same time avoiding audible recovery errors.

Figure 6.2(a) gives a schematic overview of perceptual compressed sensing in *feedforward mode*. First, from the measurement $\mathbf{y}$, the masking threshold $\mathbf{t}$ is estimated through the use of a perceptual model. Second, the sparse signal recovery step uses the measurement $\mathbf{y}$ in conjunction with the masking threshold $\mathbf{t}$ in order to recover the signal $\mathbf{s}'$.

In practice, calculating the masking threshold $\mathbf{t}$ based on the measurement $\mathbf{y}$, may not yield an accurate estimate of the masking threshold of the original signal $\mathbf{s}$. Therefore, an alternative *feedback mode* of perceptual compressed sensing is proposed and illustrated in Figure 6.2(b). In feedback mode, the masking threshold $\mathbf{t}_m$ for a current frame $\mathbf{y}_m$ is computed from the declipped signal $\mathbf{s}'_{m-1}$ of the previous frame. In the declipping algorithm proposed in subsection 6.3.3, perceptual compressed sensing will be used in feedback mode. The implications of this feedback masking threshold estimation procedure on the declipping performance will be evaluated in Subsection 6.4.3.

6.3.2 Masking Threshold Calculation

The instantaneous global masking threshold $\mathbf{t}$ of a given audio signal is calculated using part of the ISO/IEC 11172-3 MPEG-1 Layer 1 psychoacoustic model 1. A complete description of the operation of this psychoacoustic model is beyond the scope of this paper (we refer the reader to [26] and [27]). We outline the relevant steps in the computation of the instantaneous global masking threshold and illustrate the result of each step on an example audio signal (see Figure 6.3):

1. *Spectral analysis and SPL normalization*: In this step a high-resolution spectral estimate of the audio signal is calculated, with spectral components expressed in terms of sound pressure level (SPL). After a normalization operation and a Hann windowing operation on the input signal frame, the PSD estimate is obtained through a 512-point DFT. Figure 6.3(a) shows the time-domain input signal, Figure 6.3(b) shows the

resulting spectral estimate.

2. *Identification of tonal and non-tonal maskers*: It is known from psychoacoustic research that the tonality of a masking component has an influence on its masking properties [28]. For this reason it is important to discriminate between tonal maskers (defined as local maxima of the signal spectrum) and non-tonal maskers. The output of the DFT is used to determine the relevant tonal and non-tonal maskers in the spectrum of the audio signal. In a first phase, tonal maskers are identified at local maxima of the PSD: energy from three adjacent spectral components centered at the local maximum is combined to form a single tonal masker. In a second phase, a single non-tonal masker per critical band is formed by addition of all the energy from the spectral components within the critical band that have not contributed to a tonal masker.
3. *Decimation of maskers*: In this step, the number of maskers is reduced using two criteria. First, any tonal or non-tonal masker below the absolute threshold of hearing is discarded. Next, any pair of maskers occurring within a distance of 0.5 Bark is replaced by the stronger of the two. Figures 6.3(c) and 6.3(d) depict the identified tonal and non-tonal maskers respectively, after decimation.
4. *Calculation of individual masking thresholds*: an individual masking threshold is calculated for each masker in the decimated set of tonal and non-tonal maskers, using fixed psychoacoustic rules. Essentially, the individual masking threshold depends on the frequency, loudness level and tonality of the masker. Figure 6.3(e) and 6.3(f) show the individual masking thresholds associated with tonal and non-tonal maskers, respectively.
5. *Calculation of global masking threshold*: Finally, the global masking threshold is calculated by a power-additive combination of the tonal and non-tonal individual masking thresholds, and the absolute threshold of hearing. This is illustrated in Figure 6.3(g).

6.3.3 PCS-Based Declipping Using ℓ_1-norm Optimization

In order to focus CS-based declipping on the recovery of perceptually relevant signal components while at the same time avoiding audible recovery errors, the masking threshold will be incorporated into the CS recovery step. The proposed approach is to introduce in the optimization problem a diagonal *perceptual weighting matrix* where the diagonal elements are the reciprocal of the masking threshold, hence indicating the relative perceptual importance of the different signal components. The perceptual weighting matrix $\mathbf{P}_m \in \mathbb{R}^{N \times N}$ is defined as

$$\mathbf{P}_m = \begin{bmatrix} t_{m,0}^{-1} & 0 & 0 & \dots & 0 \\ 0 & t_{m,1}^{-1} & 0 & \dots & 0 \\ 0 & 0 & t_{m,2}^{-1} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & t_{m,N-1}^{-1} \end{bmatrix} \tag{6.17}$$

where $t_{m,l}$ denotes the l-th component of the m-th frame's global masking threshold $\mathbf{t}_m$.

Different ways of incorporating the perceptual weighting matrix $\mathbf{P}_m$ into the declipping optimization problem can be envisaged. The proposed way of incorporating $\mathbf{P}_m$ in the declipping of the m-th frame using ℓ_1-norm minimization, is to solve

$$\mathbf{x}'_m = \arg\min_{\mathbf{z}} \|\mathbf{P}_m\mathbf{z}\|_1 + \frac{1}{2\gamma_m}\|\mathbf{y}_m - \mathbf{A}_m\mathbf{z}\|_2^2 \tag{6.18}$$
$$\text{s.t.} \begin{cases} \mathbf{C}_m^+\mathbf{\Psi}\mathbf{z} \geq \theta_c \\ \mathbf{C}_m^-\mathbf{\Psi}\mathbf{z} \leq -\theta_c. \end{cases}$$

The declipped signal $\mathbf{s}'_m$ is constructed using

$$\begin{cases} \mathbf{C}_m^+\mathbf{s}'_m = \mathbf{C}_m^+\mathbf{\Psi}\mathbf{x}'_m \\ \mathbf{C}_m^-\mathbf{s}'_m = \mathbf{C}_m^-\mathbf{\Psi}\mathbf{x}'_m \\ \mathbf{\Phi}_m\mathbf{s}'_m = \mathbf{y}_m. \end{cases} \tag{6.19}$$

In formulation (6.18), the objective function term $\|\mathbf{P}_m\mathbf{z}\|_1$ is introduced in order to obtain a perceptually meaningful reconstruction. The perceptual weighting matrix $\mathbf{P}_m$ favors the use of those frequency components l that have a high masking threshold $t_{m,l}$. This approach is perceptually desirable:

- The introduction of distinctively audible new signal components (low masking threshold) that are not present in the original signal, is discouraged. The introduction of less audible or inaudible additional signal components (high masking threshold) is tolerated to a greater extent. As mentioned before, the masking threshold for a given frequency component indeed quantifies the signal energy for that frequency that can be masked by the original signal, or equivalently, the required signal energy for that frequency component to become audible in the simultaneous presence of the original signal.
- The recovery of perceptually important signal components present in the original signal, is encouraged. These salient signal components will, by their relatively large signal energy, indeed possess high corresponding masking thresholds.

Algorithm 8 (PCSL1)

```
Input $\mathbf{y}_m \in \mathbb{R}^M$, $\mathbf{s}'_{m-1} \in \mathbb{R}^N$, $\boldsymbol{\Psi} \in \mathbb{C}^{N \times N}$, $\theta_c \in \mathbb{R}_+$, $\gamma_m \in \mathbb{R}_+$
Output $\mathbf{s'_m} \in \mathbb{R}^N$
  $\mathbb{S}_m^r = \{l||y_{m,l}| < \theta_c\}$
  $\mathbb{S}_m^+ = \{l|y_{m,l} = \theta_c\}$
  $\mathbb{S}_m^- = \{l|y_{m,l} = -\theta_c\}$
  $\boldsymbol{\Phi}_m = I_N(\mathbb{S}_m^r)$
  $\mathbf{C}_m^+ = I_N(\mathbb{S}_m^+)$
  $\mathbf{C}_m^- = I_N(\mathbb{S}_m^-)$
  $\mathbf{A}_m = \boldsymbol{\Phi}_m \boldsymbol{\Psi}$
  Calculate $\mathbf{t}_m$ based on $\mathbf{s}'_{m-1}$ [using MPEG-1 Layer 1 psychoacoustic model 1]
  Determine $\mathbf{P}_m$ using (6.17)
  Recover $\mathbf{s}'_m$ by solving (6.18) and evaluating (6.19)
```

The perceptual weighting that is applied to the components of the sparse decomposition can be alternatively interpreted in the framework of Bayesian Compressive Sensing [29]. Omitting the constraints, the optimization problem in (6.18) is seen to be equivalent to a maximum a posteriori (MAP) formulation using independent Laplace priors [30] for each basis coefficient in $\mathbf{z}$, with mean prior values scaled by the corresponding diagonal elements of the perceptual weighting matrix $\mathbf{P}_m$.

Algorithm 8 describes the different steps of the proposed PCSL1 algorithm for declipping the m-th frame of a clipped audio signal, using ℓ_1-norm type reconstruction.

6.4 Evaluation

In order to comparatively evaluate the designed declipping algorithm with respect to existing declipping algorithms, objective tests as well as subjective listening tests have been conducted. The set-up, results and interpretation of the conducted objective and subjective experiments will be discussed in this section.

6.4.1 Objective Evaluation

To evaluate the declipping algorithms, two objective measures are used. A first measure ΔSNR indicates the physical declipping improvement and is defined

In Algorithm 8, the notation $I_N(\mathbb{S})$ is introduced to denote a matrix consisting of those rows of an $N \times N$ identity matrix I_N corresponding to the row indices in the index set $\mathbb{S}$.

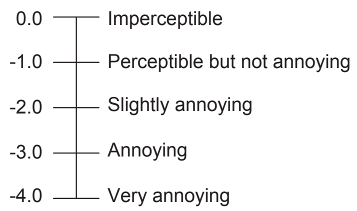


Figure 6.4: The ITU-R five-grade impairment scale

as the SNR improvement,

$$\Delta\text{SNR} = \text{SNR}(\mathbf{s}, \mathbf{s}') - \text{SNR}(\mathbf{s}, \mathbf{s_c}) \tag{6.20}$$

where

$$\text{SNR}(\mathbf{u}, \mathbf{v}) = 10 \log_{10} \left(\frac{\|\mathbf{u}\|_2^2}{\|\mathbf{u} - \mathbf{v}\|_2^2} \right) \tag{6.21}$$

and $\mathbf{s}$ is the original signal, $\mathbf{s_c}$ is the clipped signal and $\mathbf{s}'$ is the declipped signal. A second measure ΔODG indicates the perceptual declipping improvement and is defined as the Objective Difference Grade (ODG) improvement,

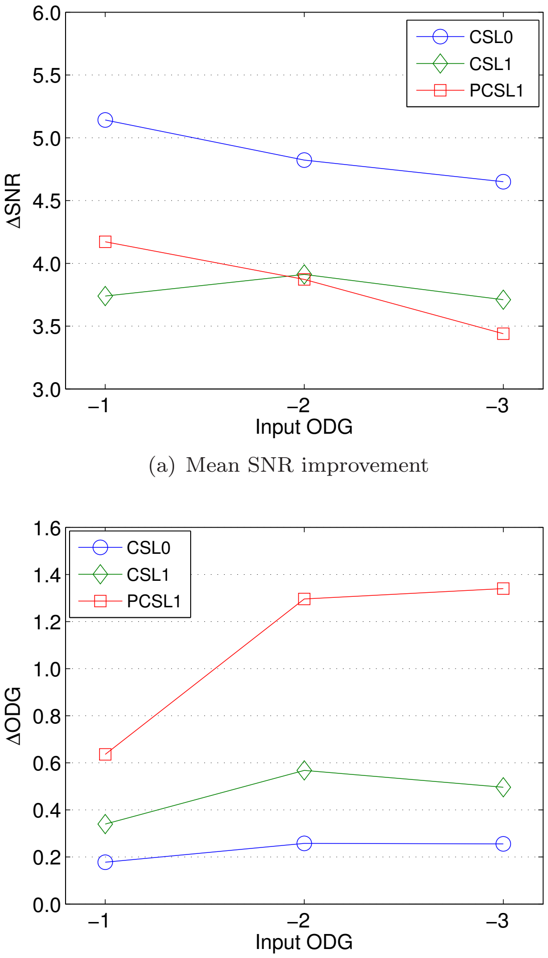
$$\Delta\text{ODG} = \text{ODG}(\mathbf{s}, \mathbf{s}') - \text{ODG}(\mathbf{s}, \mathbf{s_c}) \tag{6.22}$$

where $\text{ODG}(\mathbf{u}, \mathbf{v})$ is an objective measure of audio quality, which is calculated using the Basic Version of the PEAQ (*Perceptual Evaluation of Audio Quality*) recommendation [31]. The $\text{ODG}(\mathbf{u}, \mathbf{v})$ predicts the basic audio quality of a signal under test $\mathbf{v}$ with respect to a reference signal $\mathbf{u}$, and has a range between 0 and -4, corresponding to the ITU-R five grade impairment scale depicted in Figure 6.4. The suitability of the PEAQ ODG as an objective measure of perceived clipping degradation will be subjectively evaluated in Subsection 6.4.5.

A test database consisting of five audio excerpts was composed (16 bit mono, 44.1 kHz). The audio signals were selected so as to cover different music styles, signal dynamics and signal sparsities. The length of each audio excerpt was 10 seconds. All audio signals were normalized to the same average loudness level. Table 6.1 gives details for the different audio excerpts, including their approximate sparsity k' for the DFT basis.

Each audio signal in the test database was first clipped at three distinct clipping levels θ_c, corresponding to fixed input ODGs $\{-1, -2, -3\}$. Each clipped audio signal was subsequently processed by three different declipping algorithms, all

The approximate sparsity k' is defined as the average per-frame number of DFT signal components exceeding 0.001 times the maximal per-frame DFT signal component in absolute value, for frame length $N = 512$.



(a) Mean SNR improvement

(b) Mean ODG improvement

Figure 6.5: Comparative evaluation of different declipping algorithms in terms of objective audio quality: (a) mean ΔSNR (b) mean ΔODG scores.

			SNR($\mathbf{s}, \mathbf{s_c}$) [dB]			
Audio excerpt	Dynamics	k'	ODG=-1	ODG=-2	ODG= -3	Origin
BachPartita	high	63	32.0	24.4	22.5	[32]
FolkMusic	low	178	18.7	15.2	12.2	[33]
Mariana	low	111	45.9	42.0	37.9	[34]
Vivaldi	low	235	22.0	17.2	12.8	[35]
Whispers	high	291	16.7	12.6	9.6	[36]

Table 6.1: Audio excerpts used for the comparative evaluation of declipping algorithms

Audio excerpt	Declipping algorithm	ΔSNR/ΔODG					
		ODG=-1		ODG=-2		ODG=-3	
BachPartita	CSL0	6.79	0.34	5.24	1.09	5.60	0.43
	CSL1	6.35	-0.09	7.73	-0.03	7.92	0.16
	PCSL1	7.21	0.24	6.81	1.03	5.85	1.83
Vivaldi	CSL0	6.55	0.06	6.79	0.06	6.42	0.28
	CSL1	3.27	0.38	2.91	0.53	2.47	0.39
	PCSL1	2.98	0.81	2.63	1.33	2.26	1.32
Mariana	CSL0	4.88	0.42	4.97	0.61	4.36	0.36
	CSL1	3.65	0.63	3.70	1.14	3.31	1.03
	PCSL1	5.58	0.93	5.07	1.69	4.48	1.64
FolkMusic	CSL0	4.94	0.61	4.71	0.33	4.65	0.38
	CSL1	3.32	0.46	3.10	0.62	2.86	0.50
	PCSL1	3.14	0.41	2.95	0.63	2.76	0.68
Whispers	CSL0	2.55	-0.54	2.40	-0.80	2.22	-0.17
	CSL1	2.11	0.32	2.12	0.58	1.99	0.40
	PCSL1	1.95	0.47	1.90	1.23	1.85	0.95

Table 6.2: Objective evaluation results for different declipping algorithms.

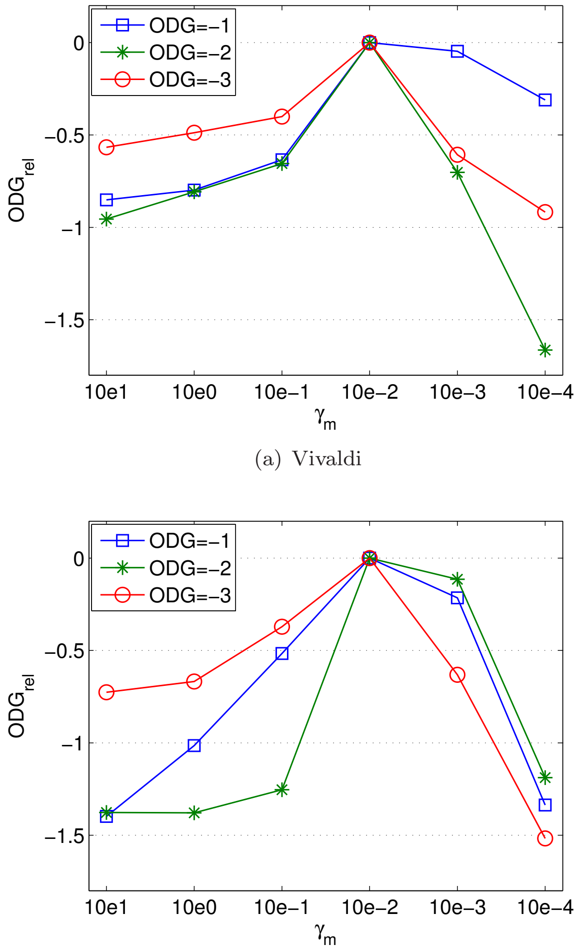
using the IDFT matrix as fixed basis $\mathbf{\Psi}$ and operating on frames of $N = 512$ samples:

- CSL0 algorithm: CS-based declipping using ℓ_0-norm optimization, all optimization problems (6.15) solved by Orthogonal Matching Pursuit [16], $\epsilon_m = 10^{-5}$. Although there are some differences (regarding the choice of the fixed basis and the choice of the stopping criterion for the OMP method), this declipping algorithm is in essence similar to the declipping algorithm proposed in [7], and can therefore adequately represent a baseline declipping algorithm in our comparative evaluation.
- CSL1 algorithm: CS-based declipping using ℓ_1-norm optimization, all optimization problems (6.16) solved by the Basis Pursuit Denoising technique proposed in [37], $\gamma_m = 10^{-2}$.
- PCSL1 algorithm: PCS-based declipping using ℓ_1-norm optimization, all optimization problems (6.18) solved by `cvx` [38], $\gamma_m = 10^{-2}$.

For each of the resulting total of 5×3×3=45 declipped audio signals, the objective measures ΔSNR and ΔODG were calculated. In Figure 6.5, the mean ΔSNR and ΔODG scores over all five audio signals are plotted as a function of the input ODG, for all considered declipping algorithms. The detailed evaluation results per audio signal are shown in Table 6.2, from which the influence of the signal sparsity on the declipping performance can be evaluated.

From Figure 6.5(a), we observe a positive average SNR improvement ranging between 3 and 6 dB for all considered declipping algorithms. This improvement appears to remain relatively constant over the input ODG range. The proposed PCSL1 algorithm does not outperform the CS-based algorithms in terms of SNR improvement. From Figure 6.5(b), the average ODG improvement is significant for all declipping algorithms. The proposed PCSL1 algorithm significantly outperforms the CS-based algorithms in terms of ODG improvement, and this for all considered input ODGs. Also note that from Table 6.2, it is observed that the best declipping performance is obtained for the sparser audio signals, such as BachPartita and Mariana.

Moreover, it is interesting to note from Figure 6.5(a) the superior performance in terms of SNR improvement of the CSL0 algorithm compared to the CSL1 algorithm. This seems to confirm an earlier suggestion [7] that, in terms of SNR improvement, the use of ℓ_0-norm optimization for declipping may be preferable over the use of ℓ_1-norm optimization. However, we observe from Figure 6.5(b) a superior performance in terms of ODG improvement of the CSL1 algorithm compared to the CSL0 algorithm, which would plead in favor of using ℓ_1-norm optimization for declipping, as far as audio quality is concerned.



(a) Vivaldi

(b) Whispers

Figure 6.6: Impact of the regularisation parameter γ_m in the PCSL1 declipping algorithm on the resulting relative (to the maximum) ODG score, for different input ODGs: (a) "Vivaldi" (b) "Whispers".

6.4.2 Impact of Regularisation Parameter γ_m

In [7], it was shown that the parameter ϵ_m in the CSL0 optimization formulation (6.15) should be carefully selected, as the resulting SNR improvement was seen to be very sensitive to the value of ϵ_m. Based upon these findings and some additional experiments, we have selected $\epsilon_m = 10^{-5}$ as an appropriate parameter setting for the experiments discussed in Subsection IV-A.

In order to study the impact of the regularisation parameter γ_m in the PCSL1 optimization formulation (6.18) on the resulting audio quality, the following experiment was conducted. Each of the five audio signals in the test database detailed in Table 6.1 were clipped at three distinct clipping levels, corresponding to fixed input ODGs $\{-1,-2,-3\}$. These signals were subsequently processed by the PCSL1 declipping algorithm using optimization formulation (6.18), in which the value of the regularisation parameter γ_m was fixed over all frames within one audio signal, and this processing was repeated for six different values of $\gamma_m = \{10^1,10^0,10^{-1}, 10^{-2},10^{-3},10^{-4}\}$. For each of the resulting total of $5\times3\times6=90$ declipped audio signals, the ODG between the original signal and the declipped signal was calculated.

The results of this experiment are partly visualized in Figure 6.6, which shows the results for the audio excerpts "Vivaldi" and "Whispers". In these figures, the objective measure ODG_{rel}, defined as

$$\text{ODG}_{\text{rel}}(\gamma_m) = \text{ODG}(\gamma_m) - \max_{\gamma_m} \text{ODG}(\gamma_m) \tag{6.23}$$

is plotted as a function of γ_m, and this for different input ODGs. We observe that the choice of γ_m has a significant impact on the resulting audio quality scores after declipping and this regardless of the input ODG, so care should be taken when selecting the value of γ_m. Moreover, we observe that the different curves reach a joint maximum for $\gamma_m = 10^{-2}$. We have selected $\gamma_m = 10^{-2}$ as an experimentally validated setting for our experiments discussed in Subsection 6.4.1.

6.4.3 Impact of Masking Threshold Estimation Procedure

In Section 6.3, the estimation of the masking threshold was discussed. It was proposed to estimate the masking threshold $\mathbf{t}_m$ for a current signal frame $\mathbf{y}_m$ by computing it from the previously declipped signal frame $\mathbf{s}'_{m-1}$. The impact of this feedback masking threshold estimation procedure on the PCSL1 declipping performance has been evaluated as follows.

The audio signals "Vivaldi" and "Whispers" (see test database detailed in Table 6.1) were clipped at three distinct clipping levels, corresponding to fixed input ODGs $\{-1,-2,-3\}$. These signals were subsequently processed by the

PCSL1 declipping algorithm, in which the per-frame masking thresholds were established using two different procedures:

- Ideal masking threshold: the per-frame masking threshold was calculated using the clean signal frame $\mathbf{s}_m$. Using this *ideal* masking threshold provides an upper bound for the declipping performance of the PCSL1 algorithm.
- Estimated masking treshold: the per-frame masking threshold $\mathbf{t}_m$ was computed using the previously declipped signal frame $\mathbf{s}'_{m-1}$, as detailed in Algorithm 8.

The results of this experiment are shown in Figure 6.7. It can be observed that for both audio excerpts, the resulting ΔODG scores using the estimated masking threshold in the declipping algorithm, are very close to the upper bounds provided by using the ideal masking threshold. These results indicate that the use of the proposed masking treshold estimation procedure does not have a significant negative impact on the resulting objective audio quality scores.

6.4.4 Subjective Evaluation

Research question and hypothesis

The research question to be answered through performing a formal subjective listening test [39] is the following: "how does the perceived audio quality improvement of audio signals declipped by the proposed PCSL1 algorithm compare to that of audio signals declipped by the CSL0 and CSL1 algorithms?". The research hypothesis, that may or may not be rejected, is that the perceived audio quality improvement is identical for all three declipping algorithms.

Test subjects

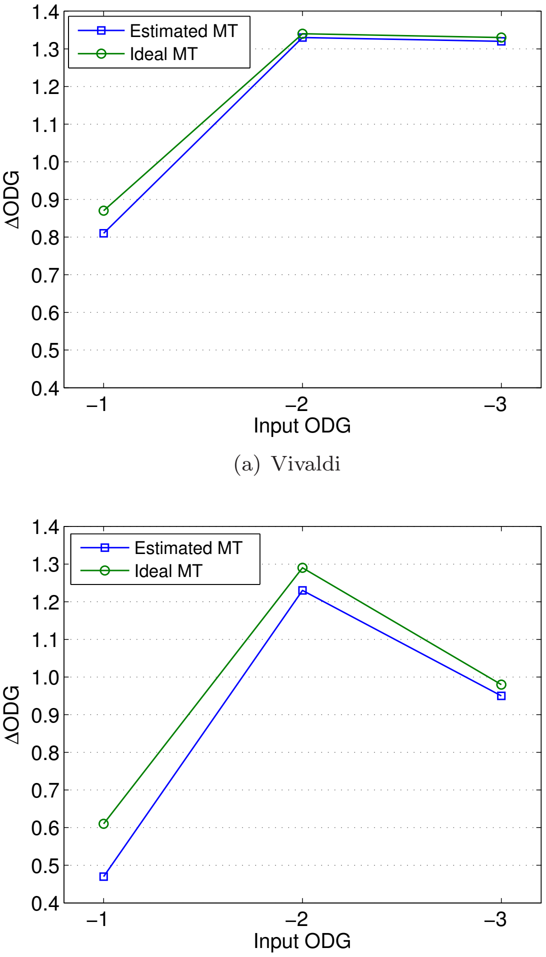
A representative sample of 16 test subjects having considerable musical listening and performance experience was selected to perform the listening test. All subjects were remunerated for their participation.

Experimental design and set-up

The listening tests were performed in a soundproof and well-illuminated test room. Stimuli were presented to the test subjects through high-quality circumaural headphones connected to a soundcard-equipped laptop. Self-developed

Sennheiser HD 439: dynamic, closed transducer, frequency response 17-22500Hz, Sound Pressure Level 112 dB, Total Harmonic Distortion $< 0.1\%$.

Sony Vaio VGN-CR41: Intel Core 2 duo T5550 processor @1.83Ghz, 3GB RAM, Realtek sound card, Intel GMA X3100 Graphics Processor.



(a) Vivaldi

(b) Whispers

Figure 6.7: Impact of the masking threshold estimation procedure in the PCSL1 declipping algorithm on the ΔODG score, for different input ODGs: (a) "Vivaldi" (b) "Whispers".

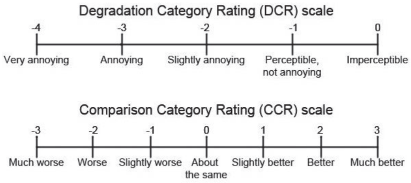


Figure 6.8: ITU-T Degradation Category Rating (DCR) and Comparison Category Rating (CCR) scales (adapted from [39]).

software was used to automate stimulus presentation and response collection. The playback level was fixed at a comfortable level.

The stimuli presented to the test subjects were the same as described in subsection 6.4.1, i.e. they consisted of the five audio excerpts detailed in Table 6.1, clipped at three distinct clipping levels, and subsequently declipped by the three declipping algorithms under study. This resulted in a total of 5×3×3=45 pairs of stimuli (each consisting of a clipped signal and the corresponding declipped signal) that were presented to the test subjects. For each pair of stimuli, the test subjects were asked to provide the following responses:

- Rate the perceived audio quality degradation of the presented clipped signal using the ITU-T Degradation Category Rating (DCR) scale [40] (see Figure 6.8).
- Rate the perceived audio quality difference of the presented declipped signal relative to the clipped signal, using the ITU-T Comparison Category Rating (CCR) scale [40] (see Figure 6.8).

Prior to the listening test, the subjects were provided with written instructions, which were verbally reviewed by the experimenter. Before the first pair of stimuli was presented, the subjects were familiarized with the effect of clipping on audio signals, by successively listening to an original sample audio signal and its clipped version. The presentation order of the pairs of stimuli was randomized using an altered Latin square scheme [39], thus eliminating possible bias effects due to order effects and sequential dependencies.

Results

The raw data resulting from the listening test consists of a categorical DCR and CCR response by each of the 16 test subjects, for each of the 45 presented pairs of stimuli. The categorical DCR and CCR responses were first converted to integers according to the scales in Figure 6.8. The analysis here will be focused on the CCR responses, the DCR responses will be used in the analysis in Subsection 6.4.5. Let us denominate the averaged CCR responses over all 16 test subjects as $\overline{\text{CCR}}$ responses, and the averaged $\overline{\text{CCR}}$ responses over all five audio excerpts as $\overline{\overline{\text{CCR}}}$ responses. In Figure 6.9, the $\overline{\overline{\text{CCR}}}$ responses are plotted as a function of the input ODG level, and this for the three different declipping algorithms under study. We observe that the ranking of the algorithms is identical to the one observed in the objective evaluation results in Figure 6.5(b). The detailed $\overline{\text{CCR}}$ responses per audio excerpt are given in Table 6.3.

The following statistical analysis was performed on the obtained numerical set of CCR responses. Let us denote the population CCR responses corresponding to audio signals declipped by the CSL0, CSL1 and PCSL1 algorithms by random variables R_{CSL0}, R_{CSL1}, and R_{PCSL1}, respectively. Based on the sample CCR responses, we tested the following three statistical hypotheses H_0 against their alternatives H_a,

$$H_0^1 \; : \; \tilde{R}_{\text{CSL0}} = \tilde{R}_{\text{PCSL1}} \tag{6.24}$$
$$H_a^1 \; : \; \tilde{R}_{\text{CSL0}} \leq \tilde{R}_{\text{PCSL1}} \tag{6.25}$$

$$H_0^2 \; : \; \tilde{R}_{\text{CSL1}} = \tilde{R}_{\text{PCSL1}} \tag{6.26}$$
$$H_a^2 \; : \; \tilde{R}_{\text{CSL1}} \leq \tilde{R}_{\text{PCSL1}} \tag{6.27}$$

$$H_0^3 \; : \; \tilde{R}_{\text{CSL0}} = \tilde{R}_{\text{CSL1}} \tag{6.28}$$
$$H_a^3 \; : \; \tilde{R}_{\text{CSL0}} \leq \tilde{R}_{\text{CSL1}} \tag{6.29}$$

where $\tilde{R}$ is the population median of the random variable R. These three statistical hypotheses were tested for all three considered input ODGs. All statistical hypotheses were tested using one-tailed Wilcoxon-Mann-Whitney tests [41] with significance level $\alpha = 0.05$. The resulting one-sided P-values are synthesized in Table 6.4.

The first null hypothesis (6.24) can be rejected in favor of the alternative (6.25) for all considered input ODGs. The second and third null hypotheses (6.26) and (6.28) can be rejected in favor of their respective alternatives (6.27) and (6.29) for input ODGs of -2 and -3.

Conclusions

The research hypothesis can be confidently rejected, i.e. the perceived audio quality improvement is not identical for all three declipping algorithms. The statistical analysis has shown that the PCSL1 algorithm delivers a significantly better audio quality improvement than both the CSL0 algorithm (for all input

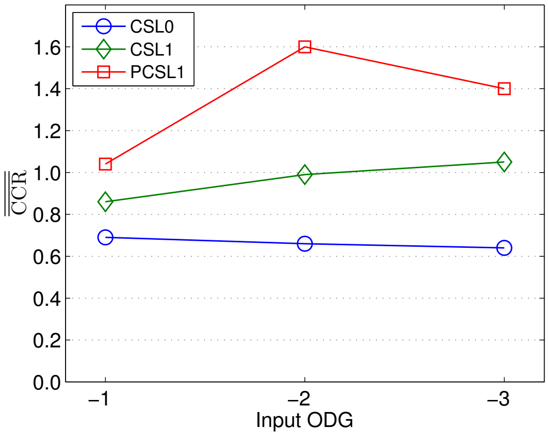


Figure 6.9: Comparative evaluation of different declipping algorithms in terms of the $\overline{\overline{\text{CCR}}}$ responses.

Audio excerpt	Declipping algorithm	$\overline{\text{CCR}}$		
		ODG=-1	ODG=-2	ODG=-3
BachPartita	CSL0	0.44	0.38	0.63
	CSL1	0.38	0.38	0.75
	PCSL1	1.63	2.56	2.31
Vivaldi	CSL0	0.25	0.75	0.44
	CSL1	0.38	0.69	0.94
	PCSL1	0.38	0.88	1.56
Mariana	CSL0	0.81	0.63	0.31
	CSL1	1.38	1.81	1.19
	PCSL1	1.69	2.19	2.06
FolkMusic	CSL0	0.63	0.44	0.75
	CSL1	0.50	1.13	1.25
	PCSL1	0.44	1.19	1.06
Whispers	CSL0	1.31	1.13	1.06
	CSL1	1.69	0.94	1.13
	PCSL1	1.06	1.19	0.88

Table 6.3: Subjective evaluation results for different declipping algorithms.

Table 6.4: P-values from one-tailed Wilcoxon-Mann-Whitney tests on sample CCR responses. Significant P-values with respect to $\alpha = 0.05$ in bold.

Null hypothesis →	H_0^1	H_0^2	H_0^3
ODG=-1	**0.0202**	0.2006	0.1313
ODG=-2	<**0.0001**	**0.0003**	**0.0178**
ODG=-3	<**0.0001**	**0.0004**	**0.0035**

ODGs) and the CSL1 algorithm (for input ODGs in {-2,-3}). Moreover, the CSL1 algorithm delivers a significantly better perceived audio quality improvement than the CSL0 algorithm (for input ODGs in {-2,-3}).

6.4.5 Suitability of PEAQ ODG as a Measure of Perceived Clipping Degradation

The PEAQ ODG measure has been developed in the context of quality evaluation of low-bit-rate encoded audio. As the nature of signal distortions introduced by clipping can be rather different as compared to signal distortions introduced by low-bit-rate codecs, the use of PEAQ ODG as an objective measure of perceived clipping-induced degradations should be well-founded. Therefore, we have investigated the appropriateness of the PEAQ ODG measure for quantifying the perceived clipping-induced audio quality degradation by analyzing the DCR data collected in the listening test described in Subsection 6.4.4.

Let us denominate the averaged DCR responses over all 16 test subjects as $\overline{\text{DCR}}$ responses. In Figure 6.10, the $\overline{\text{DCR}}$ responses are plotted as a function of the corresponding ODG score, and this for the five different audio excerpts. We observe a strong positive correlation between $\overline{\text{DCR}}$ responses and ODG scores, and this for all audio excerpts. Moreover, the different curves are seen to be monotonously increasing and they do not deviate excessively from linear curves. The different curves do have a noticeably different vertical offset, but have a nearly equal slope. These results seem to indicate that the ODG measure can be confidently used to compare the perceived clipping-induced audio quality degradation for the *same* audio excerpt in different processing scenarios (as was done in Subsection 6.4.1). However, it might not be advisable to use the ODG measure to compare the perceived clipping-induced audio quality degradation for *different* audio excerpts.

6.5 Conclusions

In this paper, a novel perceptual compressed sensing (PCS) framework has been presented for declipping audio signals, in which the theory of compressed sensing (CS) was combined with properties of human auditory perception. A declipping algorithm using ℓ_1-norm type reconstruction has been developed in the PCS framework. Comparative evaluation experiments consisting of objective and subjective tests have revealed a significant audio quality increase of the proposed PCS-based declipping algorithm compared to CS-based declipping algorithms for reasonably sparse signals.

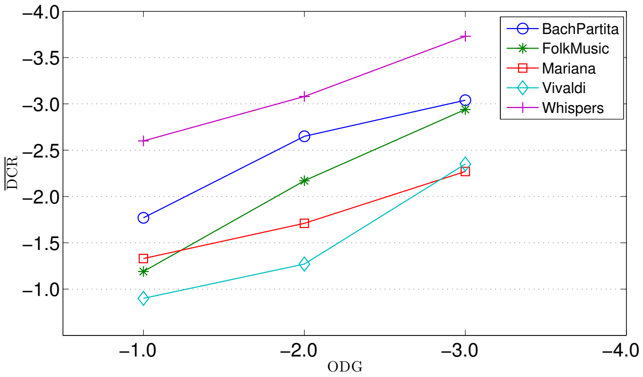


Figure 6.10: Subjective $\overline{\mathrm{DCR}}$ responses as a function of the objective ODG score.

Bibliography

[1] F. Foti, “Aliasing distortion in digital dynamics processing, the cause, effect, and method for measuring it: The story of ’digital grunge!’,” in *Preprints AES 106th Conv.*, Munich, Germany, May 1999, Preprint no. 4971.

[2] C.-T. Tan, B. C. J. Moore, and N. Zacharov, “The effect of nonlinear distortion on the perceived quality of music and speech signals,” *J. Audio Eng. Soc.*, vol. 51, no. 11, pp. 1012–1031, Nov. 2003.

[3] J. S. Abel en J. O. Smith, “Restoring a clipped signal,” in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, May 1991, pp. 1745–1748.

[4] S. J. Godsill, P. J. Wolfe, and W. N. Fong, “Statistical model-based approaches to audio restoration and analysis,” *J. New Music Res.*, vol. 30, no. 4, pp. 323–338, 2001.

[5] D. Navakauskas, “Artificial neural networks for the restoration of noise distorted songs and audio records,” Ph.D. dissertation, Vilnius Gedimina Technicial University, 1999.

[6] A. Adler, V. Emiya, M. G. Jafari, M. Elad, R. Gribonval, and M. D. Plumbley, “Audio inpainting: problem statement, relation with sparse representations and some experiments,” *Acoustics, Speech and Signal Processing, IEEE International Conference on (ICASSP 2011)*, 2011.

[7] ——, “Audio inpainting,” *IEEE Transactions on Audio, Speech and Language Processing*, 2011.

[8] A. J. Weinstein and M. B. Waking, “Recovering a clipped signal in sparseland,” *Sampling Theory in Signal and Image Processing*, to appear.

[9] C. Studer, P. Kuppinger, G. Pope, and H. Bölcskei, “Recovery of sparsely corrupted signals,” *IEEE Transactions on Information Theory*, vol. 58, no. 5, pp. 3115–3130, May 2012.

[10] M. Christensen and B. Sturm, “A perceptually reweighted mixed-norm method for sparse approximation of audio signals,” in *2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, Nov. 2011, pp. 575 –579.

[11] B. Defraene, T. van Waterschoot, H. J. Ferreau, M. Diehl, and M. Moonen, “Real-time perception-based clipping of audio signals using convex optimization,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 10, pp. 2657 –2671, Dec. 2012.

[12] E. Candes, J. Romberg, and T. Tao, “Robust uncertainty principles : Exact signal reconstruction from highly incomplete frequency information,” *IEEE Trans. on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[13] E. Candes and T. Tao, “Near optimal signal recovery from random projections: Universal encoding strategies?” *IEEE Trans. Inform. Theory*, vol. 52, no. 12, p. 54065425, 2006.

[14] D. Donoho, “Compressed sensing,” *IEEE Trans. on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[15] M. D. Plumbley, T. Blumensath, L. Daudet, R. Gribonval, and M. E. Davies, “Sparse representations in audio and music: from coding to source separation,” *Proceedings of the IEEE*, vol. 98, no. 6, pp. 995–1005, 2010.

[16] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, “Orhogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition,” *In Proc. Annual Asilomar Conference on signals, systems and computers*, 1993.

[17] S. S. Chen, D. L. Donoho, and M. A. Saunders, “Atomic decomposition by basis pursuit,” *SIAM journal on scientific computing*, vol. 20, no. 1, pp. 33–61, 1998.

[18] E. Candes and T. Tao, “The dantzig selector: Statistical estimation when p is much larger than n,” *The Annals of Statistics*, vol. 35, no. 6, pp. 2313–2351, 2007.

[19] E. Candes, J. Romberg, and T. Tao, “Stable signal recovery from incomplete and inaccurate measurements,” *Communications on pure and applied mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[20] Y. C. Eldar and G. Kutyniok, *Compressed sensing: theory and applications.* Cambridge University Press, 2012.

[21] D. L. Donoho and M. Elad, “Optimally sparse representation in general (nonorthogonal) dictionaries via l1 minimization,” *Proceedings of the National Academy of Sciences*, vol. 100, no. 5, pp. 2197–2202, 2003.

[22] L. Gan, C. Lingy, T. Doz, and T. Tranz, “Analysis of the statistical restricted isometry property for deterministic sensing matrices using steins method,” *[Online]*, http://dsp.rice.edu/?les/cs/GanStatRIP.pdf 2009.

[23] M. Rudelson and R. Vershynin, “On sparse reconstruction from fourier and gaussian measurements,” *Communications on Pure and Applied Mathematics*, vol. 61, no. 8, pp. 1025–1045, 2008.

[24] J. Yang and Y. Zhang, “Alternating direction algorithms for ℓ_1-problems in compressive sensing,” *SIAM journal on scientific computing*, vol. 33, no. 1, pp. 250–278, 2011.

[25] E. Terhardt, “Calculating virtual pitch,” *Hearing Res.*, vol. 1, no. 2, pp. 155 – 182, 1979.

[26] T. Painter and A. Spanias, “Perceptual coding of digital audio,” *Proc. IEEE*, vol. 88, no. 4, pp. 451–515, Apr. 2000.

[27] ISO/IEC, “11172-3 Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s - Part 3: Audio,” 1993.

[28] E. Zwicker and H. Fastl, *Psychoacoustics: facts and models*, 2nd ed. Springer, 1999.

[29] S. Ji, Y. Xue, and L. Carin, “Bayesian compressive sensing,” *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2346–2356, 2008.

[30] S. D. Babacan, R. Molina, and A. K. Katsaggelos, “Bayesian compressive sensing using laplace priors,” *IEEE Transactions on Image Processing*, vol. 19, no. 1, pp. 53–63, 2010.

[31] International Telecommunications Union Recommendation BS.1387, “Method for objective measurements of perceived audio quality,” 1998.

[32] J. S. Bach, “Partita no. 3 in a minor(1),” bWV 826.

[33] Naragonia, “Martin-Pecheur & Pink Molly,” Tandem.

[34] August Burns Red, “Marianas Trench,” Constellations, Solid State Records.

[35] A. Vivaldi, “The Four Seasons - Spring,” rV 269.

[36] All That Remains, “Whispers (I Hear Your),” The Fall of Ideals, Prosthetic Records.

[37] K. M. Cheman, “Optimization techniques for solving basis pursuit problems,” Ph.D. dissertation, North Carolina State University, 2006.

[38] M. Grant, S. Boyd, and Y. Ye, “Cvx: Matlab software for disciplined convex programming,” *Online accessible: http://stanford. edu/˜ boyd/cvx*, 2008.

[39] S. Bech and N. Zacharov, *Perceptual audio evaluation-Theory, method and application*. Wiley, 2007.

[40] International Telecommunications Union Recommendation P. 800, “Methods for subjective determination of transmission quality,” 1996.

[41] F. Wilcoxon, “Individual comparisons by ranking methods,” *Biometrics bulletin*, vol. 1, no. 6, pp. 80–83, 1945.

Chapter 7

Multi-Microphone Dereverberation

Embedded Optimization Algorithms for Multi-Microphone Dereverberation

Toon van Waterschoot, Bruno Defraene, Moritz Diehl, and Marc Moonen

Published in *Proc. of the 21st European Signal Process. Conf. (EUSIPCO '13)*, Marrakech, Morocco, Sept. 2013.

Contributions of second author

- literature review
- co-analysis of multi-microphone dereverberation problem
- co-development of embedded optimization problem formulation
- co-design of evaluation experiments
- co-interpretation of simulation results
- co-formulation of conclusions
- text co-editing and co-revision

Abstract

In this paper we propose a new approach to multi-microphone dereverberation, based on the recent paradigm of embedded optimization. The rationale of embedded optimization in performing online signal processing tasks is to replace traditional adaptive filtering algorithms based on closed-form estimators by fast numerical algorithms solving constrained and potentially non-convex optimization problems. In the context of dereverberation, we adopt the embedded optimization paradigm to arrive at a joint estimation of the source signal of interest and the unknown room acoustics. It is shown how the inherently non-convex joint estimation problem can be smoothed by including regularization terms based on a statistical late reverberation model and a sparsity prior for the source signal spectrum. A performance evaluation for an example multi-microphone dereverberation scenario shows promising results, thus motivating future research in this direction.

7.1 Introduction

Dereverberation refers to the process of removing reverberation from microphone signals recorded in a reverberant room. Since reverberation often has a fundamental impact on the time-frequency signal characteristics, dereverberation has been found to be a crucial component in diverse speech and audio applications, such as hearing assistance, automatic speech recognition, voice communications, and acoustic surveillance. Despite its wide applicability, dereverberation is generally still considered one of the most challenging problems in the area of acoustic signal enhancement [1]. One of the major difficulties is that dereverberation is an *inverse problem*, i.e., one aims at inverting the room impulse response (RIR), which is typically non-minimum-phase and possibly time-varying. Furthermore, dereverberation is usually also a *blind problem*, in which both the sound source signal and the room acoustics are unknown.

The state of the art in speech dereverberation can be classified into three categories [1, Ch. 1]: (1) beamforming, (2) speech enhancement, and (3) blind system identification and inversion. Most of the existing methods rely on the use of *multiple microphones*. This is implictly the case for the beamforming approaches which are based on microphone array processing, see, e.g., [2]. Speech enhancement approaches to dereverberation have also been shown to benefit from the use of multiple microphones, e.g., for accurately estimating the late reverberant signal spectrum [3] or for enhancing a linear prediction residual by spatiotemporal averaging [4]. Finally, blind system identification is typically based on the cross-relation between different microphone signals [5], while the inversion of a non-minimum-phase system has been shown to be feasible only in the multi-channel case [6].

In this paper, a different approach to dereverberation is proposed, which does not exactly fit into one of the three categories mentioned earlier. The proposed approach is somehow related to the blind system identification and inversion approach, however, it differs in that it does not require an explicit system inversion. Indeed, the major weakness of the blind system identification and inversion approach is that the design of a (multi-channel) inverse filter often appears to be an ill-posed problem, which may be due to (near-)common zeros [7] or system identification errors [8] in the RIRs. Recent solutions to alleviate this weakness are based on modifications in the inverse filter design, such as subband inversion [9], regularization [10], and forced spectral diversity [11].

Instead, we propose to avoid an explicit system inversion by adopting a recent paradigm coined as *embedded optimization*. This paradigm is based on the observation that the field of numerical optimization has reached a degree of maturity and computational efficiency such that it can be applied to online signal processing problems that are traditionally solved using recursive implementations of "classical" estimators admitting a closed-form solution [12]. In particular, it allows to directly estimate a signal vector of interest, rather than taking a detour by designing a filter to recover a signal of interest from noisy or corrupted observations.

The outline of this paper is as follows. In Section 7.2 we propose a relevant signal model and formulate the multi-microphone dereverberation problem. In Section 7.3 we propose a number of embedded optimization algorithms for multi-microphone dereverberation. These algorithms are evaluated in Section 7.4 for a simple example scenario. Finally, Section 7.5 concludes the paper.

7.2 Problem Statement

Consider a point source emitting a sound signal $s_0(t),\ t = 1, \ldots, N$, which propagates inside a room and is picked up by M microphones at different positions. The resulting microphone signals ($m = 1, \ldots, M$) are defined as

$$y_m(t) = \mathbf{h}_{m,0}^T(t)\mathbf{s}_0(t) + e_{m,0}(t),\ t = 1, \ldots, N \tag{7.1}$$

where the length-L RIR vector $\mathbf{h}_{m,0}(t)$ from the source to the mth microphone at time t is defined as

$$\mathbf{h}_{m,0}(t) = \begin{bmatrix} h_{m,0}^{(0)}(t) & \ldots & h_{m,0}^{(L-1)}(t) \end{bmatrix}^T,\ t = 1, \ldots, N \tag{7.2}$$

the length-L source signal vector $\mathbf{s}_0(t)$ at time t is defined as

$$\mathbf{s}_0(t) = \begin{bmatrix} s_0(t) & \ldots & s_0(t - L + 1) \end{bmatrix}^T,\ t = 1, \ldots, N \tag{7.3}$$

and $e_{m,0}(t),\ t = 1, \ldots, N$, denotes measurement noise.

In this paper, we make a number of assumptions that may not be valid in realistic sound acquisition scenarios, but which will allow us (1) to focus on the core issues encountered in the dereverberation problem, postponing some practical and implementation issues to future work (see Section 7.5), and (2) to investigate and interpret the proposed algorithms' behavior only w.r.t. these core issues, disregarding the potential impact of other issues on the algorithm performance. The assumptions are the following (with $m = 1, \ldots, M$):

- microphone signals are available for the entire time window $t \in [1, N]$ under consideration;
- RIRs are time-invariant within the time window $t \in [1, N]$ under consideration, i.e., $\mathbf{h}_{m,0}(t) \equiv \mathbf{h}_{m,0}$;
- initial source signal conditions $s_0(t),\ t \leq 0$ are known (and assumed equal to zero for ease of notation);
- no measurement noise is present, i.e., $e_{m,0}(t) \equiv 0$;
- all RIRs have equal and known length $L \leq N$.

Based on these assumptions, the problem considered in this paper can be formulated as follows:

Problem 1 (Multi-microphone dereverberation) *Given a length-MN vector of microphone signals generated as*

$$\mathbf{y} = \mathbf{H}_0 \mathbf{s}_0 \tag{7.4}$$

find the best possible estimate of the length-N source signal vector $\mathbf{s}_0$. Here, with $m = 1, \ldots, M$,

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1^T & \ldots & \mathbf{y}_M^T \end{bmatrix}^T, \ \mathbf{y}_m = \begin{bmatrix} y_m(1) & \ldots & y_m(N) \end{bmatrix}^T \tag{7.5}$$

$$\mathbf{H}_0 = \begin{bmatrix} \mathbf{H}_{1,0}^T & \ldots & \mathbf{H}_{M,0}^T \end{bmatrix}^T \tag{7.6}$$

$$\mathbf{H}_{m,0} = \begin{bmatrix} h_{m,0}^{(0)} & 0 & 0 & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ h_{m,0}^{(L-1)} & \ldots & h_{m,0}^{(0)} & \ddots & 0 \\ \vdots & \ddots & \vdots & \ddots & 0 \\ 0 & \ldots & h_{m,0}^{(L-1)} & \ldots & h_{m,0}^{(0)} \end{bmatrix}_{N \times N} \tag{7.7}$$

$$\mathbf{s}_0 = \begin{bmatrix} s_0(1) & \ldots & s_0(N) \end{bmatrix}^T. \tag{7.8}$$

Since the RIRs in $\mathbf{H}_0$ as well as the source signal vector $\mathbf{s}_0$ are unknown, we define the following parameter vectors,

$$\mathbf{h} = \begin{bmatrix} \mathbf{h}_1^T & \ldots & \mathbf{h}_M^T \end{bmatrix}^T, \ \mathbf{h}_m = \begin{bmatrix} h_m^{(0)} & \ldots & h_m^{(L-1)} \end{bmatrix}^T \tag{7.9}$$

$$\mathbf{s} = \begin{bmatrix} s(1) & \dots & s(N) \end{bmatrix}^T \tag{7.10}$$

$$\mathbf{e} = \begin{bmatrix} \mathbf{e}_1^T & \dots & \mathbf{e}_M^T \end{bmatrix}^T, \ \mathbf{e}_m = \begin{bmatrix} e_m(1) & \dots & e_m(N) \end{bmatrix}^T \tag{7.11}$$

and a data model admitting two equivalent formulations,

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{e} \tag{7.12}$$

$$= (\mathbf{I}_M \otimes \mathbf{S})\,\mathbf{h} + \mathbf{e} \tag{7.13}$$

where $\mathbf{H}$ is a $MN \times N$ matrix with the coefficients of the RIRs parameter vector $\mathbf{h}$ in a block Toeplitz structure as in (7.6)-(7.7), $\mathbf{I}_M$ is the $M \times M$ identity matrix, $\otimes$ denotes the Kronecker product, and $\mathbf{S}$ is a $N \times L$ Toeplitz matrix defined as

$$\mathbf{S} = \begin{bmatrix} s(1) & \dots & 0 \\ \vdots & \ddots & \vdots \\ s(N-L+1) & \dots & s(1) \\ \vdots & \ddots & \vdots \\ s(N) & \dots & s(N-L+1) \end{bmatrix}. \tag{7.14}$$

The error signal parameter vector $\mathbf{e}$ is included to account for estimation errors in both $\mathbf{h}$ and $\mathbf{s}$.

7.3 Embedded Optimization Algorithms

State-of-the-art multi-microphone dereverberation algorithms in the category of blind system identification and inversion approach Problem 1 using a two-step procedure. First, an estimate $\hat{\mathbf{H}}$ of the RIRs matrix is computed using a blind identification method that typically exploits the cross-relation between different microphone signals [5]. Second, an M-input, single-output inverse filter $\mathbf{g}$ is designed and an estimate of the source signal vector is obtained as

$$\hat{\mathbf{s}} = \mathbf{G}\mathbf{y} \tag{7.15}$$

where $\mathbf{G}$ is a block Toeplitz matrix of appropriate dimensions, containing the inverse filter coefficients in $\mathbf{g}$.

Instead, we propose to *jointly* estimate the RIRs parameter vector $\mathbf{h}$ and the source signal parameter vector $\mathbf{s}$. We derive three nonlinear least squares (NLS) optimization problems for estimating $\mathbf{h}$ and $\mathbf{s}$, and point out their strengths and weaknesses. More particularly, we consider NLS problems without regularization (**NLS**), with ℓ_2-norm regularization exploiting prior knowledge on $\mathbf{h}$ (**$\boldsymbol{\ell_2}$-RNLS**), and with ℓ_1-norm and ℓ_2-norm regularization exploiting prior knowledge on $\mathbf{s}$ and $\mathbf{h}$ (**$\boldsymbol{\ell_1/\ell_2}$-RNLS**). A block coordinate descent (BCD) approach is adopted for solving these problems, resulting in three iterative algorithms in which $\mathbf{h}$ and $\mathbf{s}$ are estimated sequentially.

The sequential nature of the proposed algorithms shows a certain degree of similarity with the state-of-the-art two-step procedure for blind system identification and inversion. A crucial difference, however, is that the RIRs parameter vector $\mathbf{h}$ is not just estimated once, but its estimate is iteratively refined as improved estimates of the source signal parameter vector become available. Another similarity with the state of the art, is that the source signal parameter vector estimate resulting from the NLS and ℓ_2-RNLS problems is linearly related to the microphone signal vector $\mathbf{y}$, so that it can be interpreted as the result of an inverse filtering approach, even though an inverse filter is never explicitly designed or computed. When solving the ℓ_1/ℓ_2-RNLS problem, however, the source signal parameter vector estimate is not linearly related to the microphone signal vector and so an inverse filtering interpretation is not appropriate.

7.3.1 NLS problem

The starting point for the derivation of embedded optimization algorithms solving Problem 1, is the formulation of an NLS optimization problem for the data model (7.12)-(7.13),

$$\min_{\mathbf{h},\mathbf{s},\mathbf{e}} \quad \|\mathbf{e}\|_2^2 \tag{7.16}$$

$$\text{s. t.} \quad \mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{e} \tag{7.17}$$

$$= (\mathbf{I}_M \otimes \mathbf{S})\,\mathbf{h} + \mathbf{e}. \tag{7.18}$$

The proposed solution strategy consists in first minimizing (7.16) w.r.t. $\{\mathbf{s}, \mathbf{e}\}$ for a fixed value of $\mathbf{h} = \hat{\mathbf{h}}$ using the equality constraints in (7.17), then minimizing (7.16) w.r.t. $\{\mathbf{h}, \mathbf{e}\}$ for a fixed value of $\mathbf{s} = \hat{\mathbf{s}}$ using the equality constraints in (7.18), and repeating this procedure for a number of iterations (here fixed to $k_{\max}$). The resulting BCD algorithm is shown in Algorithm 9, where $(\cdot)^+$ denotes the Moore-Penrose pseudoinverse.

Algorithm 9 BCD algorithm for NLS problem

```
Input initial RIRs parameter vector estimate $\hat{\mathbf{h}}^{(0)}$
Output parameter vector estimates $\hat{\mathbf{s}} = \hat{\mathbf{s}}^{(k_{\max})}, \hat{\mathbf{h}} = \hat{\mathbf{h}}^{(k_{\max})}$
for $k = 1, \ldots, k_{\max}$ do
    $\hat{\mathbf{s}}^{(k)} = (\hat{\mathbf{H}}^{(k-1)})^+\mathbf{y}$
    $\hat{\mathbf{h}}^{(k)} = \left(\mathbf{I}_M \otimes (\hat{\mathbf{S}}^{(k)})^+\right)\mathbf{y}$
end for
```

7.3.2 ℓ_2-regularized NLS problem

It is well known that the NLS optimization problem in (7.16)-(7.18) generally has multiple local solutions, and the BCD algorithm will only converge to the global solution if the algorithm is properly initialized (i.e., if a good initial estimate for either $\mathbf{h}$ or $\mathbf{s}$ is available). An effective approach for smoothing the NLS objective function, and hence facilitating convergence to a meaningful local solution, is the addition of a regularization term incorporating prior knowledge on the unknown parameter vectors. A first approach to regularization consists in the addition of a weighted ℓ_2-norm of the RIRs parameter vector $\mathbf{h}$ to (7.16),

$$\begin{aligned} \min_{\mathbf{h},\mathbf{s},\mathbf{e}} \quad & \|\mathbf{e}\|_2^2 + \lambda_1 \|\mathbf{h}\|_{\mathbf{W}}^2 && (7.19) \\ \text{s. t.} \quad & \mathbf{y} = \mathbf{Hs} + \mathbf{e} && (7.20) \\ & \quad = (\mathbf{I}_M \otimes \mathbf{S})\,\mathbf{h} + \mathbf{e} && (7.21) \end{aligned}$$

A mean-square-error optimal choice for the weighting matrix $\mathbf{W}$ corresponds to the inverse covariance matrix of the true RIRs vector $\mathbf{h}_0 = \begin{bmatrix} \mathbf{h}_{1,0}^T & \dots & \mathbf{h}_{M,0}^T \end{bmatrix}^T$, which is considered to be a random zero-mean variable having a Gaussian probability density function. In the context of dereverberation, the use of a statistical model for the late reverberation component in the RIRs has been proven useful in a variety of algorithms. The most commonly used model is the so-called Polack's model [3], which approximates the temporal envelope of the RIRs using an exponential function with a fixed decay α. We will adopt this model in the proposed algorithm, and neglect any cross-correlations between the RIRs parameters, such that $\mathbf{W}$ can be defined as a diagonal matrix,

$$\mathbf{W} = \mathbf{I}_M \otimes \operatorname{diag}\left\{1, e^{2\alpha}, \dots, e^{2(L-1)\alpha}\right\}. \quad (7.22)$$

The resulting BCD algorithm is given in Algorithm 10.

Algorithm 10 BCD algorithm for ℓ_2-RNLS problem

```
Input initial RIRs parameter vector estimate $\hat{\mathbf{h}}^{(0)}$, Polack's model decay $\alpha$,
    regularization parameter $\lambda_1$
Output parameter vector estimates $\hat{\mathbf{s}} = \hat{\mathbf{s}}^{(k_{\max})}, \hat{\mathbf{h}} = \hat{\mathbf{h}}^{(k_{\max})}$
$\bar{\mathbf{W}} = \operatorname{diag}\{1, e^{2\alpha}, \dots, e^{2(L-1)\alpha}\}$
for $k = 1, \dots, k_{\max}$ do
    $\hat{\mathbf{s}}^{(k)} = (\hat{\mathbf{H}}^{(k-1)})^{+}\mathbf{y}$
    $\hat{\mathbf{h}}^{(k)} = \left(\mathbf{I}_M \otimes \left[(\hat{\mathbf{S}}^{(k)T}\hat{\mathbf{S}}^{(k)} + \lambda_1\bar{\mathbf{W}})^{-1}\hat{\mathbf{S}}^{(k)T}\right]\right)\mathbf{y}$
end for
```

7.3.3 ℓ_1/ℓ_2-regularized NLS problem

With the aim of obtaining an additional smoothing effect, prior knowledge on the source signal vector can also be incorporated in the ℓ_2-RNLS optimization problem. Building on the proven efficiency of sparse representations for speech and audio signals [13], an ℓ_1-norm regularization in a suitable spectral basis seems to be appropriate for this purpose. When combined with the ℓ_2-norm regularization of the RIRs parameter vector, this results in the following optimization problem,

$$\min_{\mathbf{h},\mathbf{s},\mathbf{e}} \quad \|\mathbf{e}\|_2^2 + \lambda_1 \|\mathbf{h}\|_{\mathbf{W}}^2 + \lambda_2 \|\mathbf{D}\mathbf{s}\|_1 \tag{7.23}$$

$$\text{s. t.} \quad \mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{e} \tag{7.24}$$

$$= (\mathbf{I}_M \otimes \mathbf{S})\,\mathbf{h} + \mathbf{e}. \tag{7.25}$$

Here, $\mathbf{D}$ is an $N \times N$ orthogonal matrix defining a spectral transform, such as the discrete Fourier or cosine transform (DFT/DCT). In contrast to the previous two problems, the optimization problem in (7.23)-(7.25) does not admit a closed-form solution when optimizing w.r.t. $\{\mathbf{s}, \mathbf{e}\}$. However, this particular subproblem is convex and can therefore be efficiently solved using existing software (e.g., we use CVX/SeDuMi). The resulting BCD algorithm is shown in Algorithm 11.

Algorithm 11 BCD algorithm for ℓ_1/ℓ_2-RNLS problem

```
Input initial RIRs parameter vector estimate $\hat{\mathbf{h}}^{(0)}$, Polack's model decay $\alpha$,
    orthogonal spectral transform matrix $\mathbf{D}$, regularization parameters $\lambda_1, \lambda_2$
Output parameter vector estimates $\hat{\mathbf{s}} = \hat{\mathbf{s}}^{(k_{\max})}, \hat{\mathbf{h}} = \hat{\mathbf{h}}^{(k_{\max})}$
$\bar{\mathbf{W}} = \mathrm{diag}\left\{1, e^{2\alpha}, \dots, e^{2(L-1)\alpha}\right\}$
for $k = 1, \dots, k_{\max}$ do
    $\hat{\mathbf{s}}^{(k)} = \arg\min_{\mathbf{s}} \|\mathbf{y} - \hat{\mathbf{H}}^{(k-1)}\mathbf{s}\|_2^2 + \lambda_2 \|\mathbf{D}\mathbf{s}\|_1$
    $\hat{\mathbf{h}}^{(k)} = \left(\mathbf{I}_M \otimes \left[(\hat{\mathbf{S}}^{(k)T}\hat{\mathbf{S}}^{(k)} + \lambda_1 \bar{\mathbf{W}})^{-1}\hat{\mathbf{S}}^{(k)T}\right]\right)\mathbf{y}$
end for
```

7.4 Evaluation

The proposed embedded optimization algorithms for multi-microphone dereverberation are evaluated here by means of a simulation example. A microphone signal vector $\mathbf{y}$ is generated by filtering a source signal vector $\mathbf{s}_0$ of length $N = 1024$, corresponding to a quasi-stationary voiced segment of a male speech signal sampled at 8 kHz, using $M = 5$ synthetic RIRs of length $L = 100$. The RIRs are synthesized by shaping $M = 5$ different Gaussian white noise (GWN) sequences with an exponential envelope corresponding to Polack's model with $\alpha = 0.025$. The same envelope is used for designing the weighting matrix $\mathbf{W}$ in the ℓ_2-RNLS and ℓ_1/ℓ_2-RNLS problems. The regularization parameters have been chosen as $\lambda_1 = \lambda_2 = 0.1$, and $\mathbf{D}$ is the DCT matrix. All algorithms start from a random GWN initial RIRs parameter vector estimate $\hat{\mathbf{h}}^{(0)}$ and perform $k_{\max} = 10$ iterations. In the simulation results, the inherent scaling ambiguity has been removed by plotting $\hat{\mathbf{s}}/a$ and $a\hat{\mathbf{h}}$ rather than $\hat{\mathbf{s}}$ and $\hat{\mathbf{h}}$, with $a = \sqrt{\hat{\mathbf{s}}^T\hat{\mathbf{s}}/\mathbf{s}_0^T\mathbf{s}_0}$.

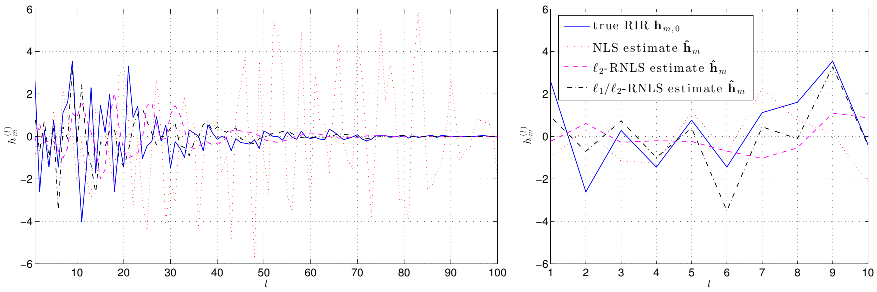
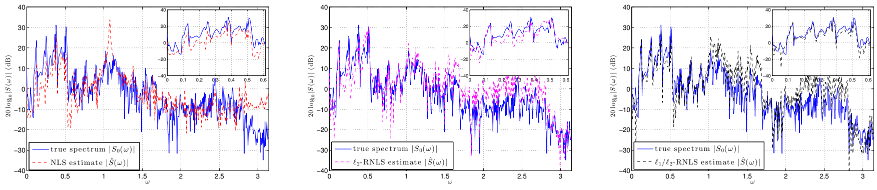


Figure 7.1: Comparison of true RIR and RIR parameter vector estimates for $m = 2$.



(a) NLS

(b) ℓ_2-RNLS

(c) ℓ_1/ℓ_2-RNLS

Figure 7.2: Comparison of magnitude spectra of true source signal and source signal parameter vector estimates.

Fig. 7.1 shows the true and estimated RIR ($m = 2$), while Fig. 7.2 compares the magnitude spectrum of the true and estimated source signal. As expected, the BCD algorithm does not converge to the global NLS problem solution, and suffers from a severe overestimation of the coefficients in the RIR tail as well as large ($\geq$ 10 dB) source spectrum estimation errors in some frequency regions. The ℓ_2-norm regularization is seen to have a beneficial effect on the *overall* estimation performance, yielding RIR and source spectrum estimates that follow the envelopes of the true RIR and source spectrum. In addition, the ℓ_1-norm regularization further increases the *local* estimation performance: the right plot in Fig. 7.1 shows an improved estimation of the early RIR coefficients, while the top right subplot in Fig. 7.2(c) illustrates the improved accuracy of the estimated quasi-harmonic speech components.

7.5 Conclusions

In this paper, we have introduced a new approach to multi-microphone dereverberation, based on a recent paradigm known as embedded optimization. Three sequential optimization algorithms have been proposed, which enable the joint estimation of the unknown source signal and room acoustics. By adopting an iterative numerical optimization strategy, the need for an explicit inverse filter design is avoided. However, the inclusion of appropriate regularization terms in the inherently non-convex optimization problem appears to be crucial for assuring convergence to a meaningful local solution. In particular, the addition of a weighted ℓ_2-norm of the RIRs parameter vector, based on a statistical model for late reverberation, leads to an improved overall estimation performance. In addition, the accuracy of the estimated early reflections and (quasi-)harmonic source signal components can be further increased by incorporating an ℓ_1-norm regularization term for the source signal parameter vector DFT/DCT.

The work presented in this paper is a first step towards the development of efficient and reliable embedded optimization algorithms for multi-microphone dereverberation. A number of challenges for future research remain, e.g.,

- to move from batch to online (frame-based) processing, properly managing initial/final conditions,
- to generalize the ℓ_2-norm regularization for dealing with realistic impulse responses,
- to take measurement noise into account,
- to arrive at autonomous optimization algorithms involving proper termination criteria and cross-validation procedures for adjusting the regularization parameters,
- to derive fast SQP/SCP algorithms exploiting the particular dereverberation problem structure,
- to use perceptual criteria in the problem formulations,
- to evaluate the resulting dereverberation performance.

Bibliography

[1] P. A. Naylor and N. D. Gaubitch, Eds., *Speech dereverberation.* London: Springer, 2010.

[2] J. L. Flanagan, A. C. Surendran, and E. E. Jan, “Spatially selective sound capture for speech and audio processing,” vol. 13, no. 1–2, pp. 207–222, Oct. 1993.

[3] E. A. P. Habets, “Multi-channel speech dereverberation based on a statistical model of late reverberation,” vol. 4, Philadelphia, PA, USA, Mar. 2005, pp. 173–176.

[4] N. D. Gaubitch and P. A. Naylor, “Spatiotemporal averaging method for enhancement of reverberant speech,” Cardiff, UK, Jun. 2007, pp. 607–610.

[5] G. Xu, H. Liu, L. Tong, and T. Kailath, “A least-squares approach to blind channel identification,” vol. 43, no. 12, pp. 2982–2993, Dec. 1995.

[6] M. Miyoshi and Y. Kaneda, “Inverse filtering of room acoustics,” vol. ASSP-36, no. 2, pp. 145–152, Feb. 1988.

[7] W. Zhang and P. A. Naylor, “An experimental study of the robustness of multichannel inverse filtering systems to near-common zeros,” Glasgow, Scotland, UK, Aug. 2009, pp. 194–198.

[8] ——, “A system-identification-error-robust method for equalization of multichannel acoustic systems,” Dallas, TX, USA, Mar. 2010, pp. 109–112.

[9] N. D. Gaubitch and P. A. Naylor, “Equalization of multichannel acoustic systems in oversampled subbands,” vol. 17, no. 6, pp. 1061–1070, Aug. 2009.

[10] I. Kodrasi, S. Goetze, and S. Doclo, “Increasing the robustness of acoustic multichannel equalization by means of regularization,” Aachen, Germany, Sep. 2012, pp. 161–164.

[11] X. S. Lin, A. W. H. Khong, and P. A. Naylor, “A forced spectral diversity algorithm for speech dereverberation in the presence of near-common zeros,” vol. 20, no. 3, pp. 888–899, Mar. 2012.

[12] J. Mattingley and S. Boyd, “Real-time convex optimization in signal processing,” vol. 27, no. 3, pp. 50–61, May 2010.

[13] M. D. Plumbey, T. Blumensath, L. Daudet, R. Gribonval, and M. E. Davies, “Sparse representations in audio and music: from coding to source separation,” vol. 98, no. 6, pp. 995–1005, 2010.

Chapter 8

Conclusions and Suggestions for Future Research

8.1 Summary and Conclusions

In this thesis we have designed an embedded optimization framework for audio signal enhancement and investigated its application to four different audio signal enhancement problems, namely hard clipping precompensation, loudspeaker precompensation, declipping and multi-microphone dereverberation. The research objectives formulated in **Chapter 1** were the conceptual development of an embedded optimization framework for audio signal enchancement, the incorporation of perceptual information into this framework, the design and hardware implementation of application-tailored optimization methods, and the comparative objective and subjective evaluation with state-of-the-art audio signal enhancement algorithms. These research objectives have been achieved, resulting in the following four major contributions of this thesis.

The first major contribution consists in the conceptual development of an *embedded optimization framework* and its application to different audio signal enchancement problems. In a first part of this thesis, precompensation algorithms for audio signal enhancement have been considered. Hard clipping precompensation was formulated in **Chapter 2** as a sequence of per-frame constrained convex quadratic optimization problems. Loudspeaker precompensation was formulated in **Chapter 3** as a sequence of per-frame nonconvex optimization problems, in which the loudspeaker was modeled by a grey-box Hammerstein loudspeaker model, i.e. a cascade of a memoryless nonlinearity and a linear finite impulse response filter. In a second part of this thesis, recovery algorithms for audio signal enhancement have been considered. Declipping was formulated in **Chapter 6** as a sparse signal recovery problem in the DFT domain, in which the recovery was performed by solving a per-frame ℓ_1-norm-type convex optimization problem. Multi-microphone dereverberation was formulated in **Chapter 7** as a nonconvex optimization problem, allowing for the joint estimation of the room acoustics model parameters and the clean audio signal.

The second major contribution of this thesis consists in the incorporation of *perceptual information* into the embedded optimization algorithms through the use of a model of human sound perception. In the proposed embedded optimization precompensation algorithms, the MPEG perceptual model [1] was applied to compute the instantaneous masking threshold, in order to establish an objective mathematical measure reflecting the perceptible distortion incurred in the subsequent distortion process. In the hard clipping precompensation algorithm presented in **Chapter 2**, an objective function reflecting the perceptible nonlinear clipping distortion was constructed by including frequency weights based on the instantaneous masking threshold. In the loudspeaker precompensation algorithm presented in **Chapter 3**, an objective function reflecting the perceptible combined linear and nonlinear loudspeaker distortion was constructed in a similar fashion. In the declipping algorithm presented in **Chapter 6**, frequency weights based on the instantaneous masking threshold were included in the ℓ_1-norm optimization criterion. As a result, the declipping algorithm was focused on maximizing the perceived audio quality of the declipped signal instead of the physical signal reconstruction quality. In the multi-microphone dereverberation algorithm of **Chapter 7**, perceptual information was not explicitly incorporated into the embedded optimization algorithm. However, the addition of regularization terms based on a statistical late reverberation model and a sparsity prior for the clean audio signal spectrum has shown that the optimization problem formulation lends itself very well to the inclusion of prior information, which could be easily extended to the inclusion of perceptual information.

The third major contribution of this thesis consists in the *design and hardware implementation* of *fast and reliable optimization methods* for solving the per-frame optimization problems. As concerns hard clipping precompensation, three different structure-exploiting convex optimization methods were proposed in **Chapter 2** for solving the per-frame constrained convex quadratic optimization problems. A first method was an active set method, a second and third method were projected gradient methods, for which theoretical complexity bounds were derived. The fastest of these optimization methods was seen to be the optimal projected gradient method originally proposed in [2], which was shown to run in real time on a standard PC. The hardware implementation of the optimal projected gradient optimization method on a field programmable gate array (FPGA) was discussed **Chapter 5**. Bit-accurate simulations of the FPGA implementation have demonstrated that the transition from floating-point arithmetic to fixed-point arithmetic and the selection of the corresponding bit width has an impact on the resulting audio quality as well as on the resource usage and power consumption of the design. The selected FPGA design solves the per-frame optimization problems by performing 30 optimal projected gradient iterations using 20 fixed-point fraction bits, has a low latency, a reduced power consumption, a reduced resource usage, and at the same time preserves the full audio quality improvement. This design allows to perform the algorithm in real time on a small and portable audio device. In **Chapter 3**, the solution of the per-frame nonconvex optimization problems for loudspeaker precompensation was considered. Depending on the invertibility and the smoothness of the memoryless nonlinearity in the Hammerstein loudspeaker model, different optimization strategies were proposed. In the case of an invertible memoryless nonlinearity, the optimization problem was seen to have a closed-form solution. In the case of a non-invertible smooth memoryless nonlinearity, a gradient optimization method was applied, whereas for non-invertible hard clipping memoryless nonlinearities, the optimal projected gradient optimization method was applied. For the multi-microphone dereverberation algorithm discussed in **Chapter 7**, the resulting nonconvex joint estimation problem was solved using a block coordinate descent (BCD) approach, sequentially estimating the room acoustics model parameters and the clean audio signal.

The fourth major contribution of this thesis consists in the *comparative objective and subjective evaluation* of the designed embedded optimization algorithms with state-of-the-art audio signal enhancement algorithms. For hard clipping precompensation, the objective audio quality evaluation performed in **Chapter 2** reported significantly higher objective audio quality scores using the embedded optimization algorithm over standard hard clipping precompensation techniques, and this for moderate to high levels of clipping. For loudspeaker precompensation, the objective evaluation experiments using synthetic and identified loudspeaker models performed in **Chapter 3** have shown that the embedded optimization loudspeaker precompensation algorithm provides a significant audio quality improvement, especially so at high loudspeaker playback levels. In **Chapter 4**, a subjective evaluation of the aforementioned hard clipping and loudspeaker precompensation algorithms was performed, through a formal listening test including 19 test subjects. The conclusions of this subjective evaluation were seen to confirm the outcome of the objective evaluation experiments. Moreover, some interesting side results were obtained from the subjective evaluation experiments. First, the perceived audio quality of audio signals subject to a certain nonlinear hard clipping distortion was seen to be not significantly superior to that of audio signals subject to the same nonlinear hard clipping distortion and an additional linear distortion. This confirms earlier findings [3] that nonlinear distortion is dominant in the perception with respect to linear distortion. Second, there was a significant positive correlation between subjective audio quality scores and objective audio quality scores, supporting the validity of using objective measures to assess the audio quality of signals subject to hard clipping and loudspeaker distortion. In **Chapter 6**, objective and subjective evaluation experiments (formal listening test using 16 test subjects) have revealed a significant audio quality increase for the proposed embedded optimization declipping algorithm compared to existing CS-based declipping algorithms.

8.2 Suggestions for Future Research

In this thesis, precompensation algorithms and recovery algorithms for audio signal enhancement have been considered in an embedded optimization framework. The dichotomy between precompensation and recovery algorithms was introduced due to the observation that in many cases, the audio signal can only be processed either before or after the distortion process acts onto the audio signal. However, one can think of commonly encountered scenarios where the audio signal can be processed both before and after the action of the distortion process, e.g. a predictable hard clipping distortion process, or a distortion process occuring during audio signal mastering or transmission stages. In these scenarios, a precompensation algorithm and a recovery algorithm could cooperate in order to maximally mitigate the perceptible effects of the distortion process. This joint integration of precompensation and recovery in an embedded optimization approach is in our view a highly interesting and relevant direction for future research.

Next, the application of a model of human sound perception in the proposed embedded optimization framework was seen to result in an improved audio quality for hard clipping precompensation, loudspeaker precompensation and declipping problems. When the goal is to improve another perceptual attribute, such as speech quality or speech intelligibility, it will be necessary to incorporate a perceptual model translating these specific perceptual attributes to mathematical measures that can be optimized for. The study and integration of these perceptual models is an important step to allow the embedded optimization framework to maximally enhance these perceptual attributes. Moreover, whereas the focus of this thesis has been on improving human sound perception, it is our opinion that the embedded optimization approach is equally suited for applications where the aim is to improve the sound perception by machines, e.g. in automatic speech recognition, given the proper inclusion of models of machine sound perception.

Concerning the fast and reliable solution of the per-frame optimization problems that form the core of the embedded optimization algorithms, we have followed the path of exploiting as much as possible the inherent structure of the optimization problems. For the considered convex optimization problems, we have if possible also derived theoretical upper bounds on the algorithmic complexity. In this approach, the resulting application-specific optimization methods have to be implemented in or translated to compilable and reliable code that can be embedded in real-time audio applications. We note that a different paradigm called automatic code generation [4] can be envisaged for performing embedded optimization in real-time audio signal enhancement applications. This approach takes a high-level description of an optimization problem family, and automatically generates code that compiles into a reliable and high-speed solver for the problem family. The exploration of automatic code generation approaches for embedded optimization in audio signal enhancement applications and their comparison in terms of speed and reliability with the optimization approach presented in this thesis forms a highly relevant topic for future research.

Finally, we also want to point out that most of the proposed embedded optimization algorithms rely on an estimate of the model parameters of the distortion process under consideration. However, the distortion model parameter estimation will in general not be performed without estimation errors. It is therefore worthwile to assess the sensitivity of the resulting audio signal enhancement performance of the different embedded optimization algorithms to estimation errors in the distortion model parameters.

Beside the suggested general directions for future research discussed above, we can point out some potential research directions for the specific audio signal enhancement problems of loudspeaker compensation and multi-microphone dereverberation.

Loudspeaker Precompensation

In this thesis, the loudspeaker has been modeled using a grey-box Hammerstein model, which gives a good trade-off between model accuracy and model complexity. The application of more complex loudspeaker models, such as Volterra models or white-box lumped parameter models could result in a higher accuracy in modeling the different loudspeaker distortion mechanisms. However, the precompensation of such loudspeaker models using embedded optimization will presumably result in more complex and typically nonconvex optimization problems. The adequate formulation and efficient solution of such optimization problems involving more complex loudspeaker models, and the comparison between the precompensation performance for different loudspeaker models would form major contributions to the loudspeaker compensation problem.

Moreover, in this thesis the loudspeaker model parameters were estimated offline during a separate estimation procedure. In practice, the loudspeaker model parameters might vary over time, e.g. due to temperature changes. On-line tracking of the time variation of the loudspeaker model parameters is a challenging problem due to the difficulty to properly feed back the reproduced audio signal, which could be achieved using some form of measurement and feedback of the loudspeaker voice coil displacement or temperature.

Multi-Microphone Dereverberation

In this thesis, an embedded optimization approach to multi-microphone dereverberation has been presented, which was seen to differ fundamentally from existing dereverberation approaches. The promising results of this approach for an example multi-microphone dereverberation scenario motivate future research in this direction. In our view, the most important research topics to be addressed are the derivation of fast optimization methods exploiting the particular dereverberation problem structure, the inclusion of measurement noise, the incorporation of perceptual criteria in the problem formulation, and the evaluation of the resulting dereverberation performance using objective and subjective speech intelligibility experiments.

Bibliography

[1] ISO/IEC, “11172-3 Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s - Part 3: Audio,” 1993.

[2] Y. Nesterov, *Introductory lectures on convex optimization.* Springer, 2004.

[3] B. C. Moore, C.-T. Tan, N. Zacharov, and V.-V. Mattila, “Measuring and predicting the perceived quality of music and speech subjected to combined linear and nonlinear distortion,” *Journal of the Audio Engineering Society*, vol. 52, no. 12, pp. 1228–1244, 2004.

[4] J. Mattingley and S. Boyd, “Cvxgen: a code generator for embedded convex optimization,” *Optimization and Engineering*, vol. 13, no. 1, pp. 1–27, 2012.

Publication List

International Journal Papers

1. B. Defraene, T. van Waterschoot, H.J. Ferreau, M. Diehl, and M. Moonen, "Real-time perception-based clipping of audio signals using convex optimization," *IEEE Trans. Audio Speech Language Process.*, vol. 20, no. 10, pp. 2657–2671, Dec. 2012.

2. B. Defraene, N. Mansour, S. De Hertogh, T. van Waterschoot, M .Diehl, and M .Moonen, "Declipping of audio signals using perceptual compressed sensing," *IEEE Trans. Audio Speech Language Process.*, vol. 21, no. 12, pp. 2627-2637, Dec. 2013.

3. B. Defraene, T. van Waterschoot, M. Diehl, and M. Moonen, "Embedded-optimization-based loudspeaker compensation using a Hammerstein loudspeaker model," *IEEE Trans. Audio Speech Language Process.*, submitted for publication, Nov. 2013.

4. B. Defraene, T. van Waterschoot, M. Diehl, and M. Moonen, "Subjective evaluation of precompensation algorithms for audio signal enhancement," to be submitted for publication, Dec. 2013.

5. B. Defraene, A. Suardi, T. van Waterschoot, M. Diehl, and M. Moonen, "An FPGA implementation of the perception-based clipping algorithm,", to be submitted for publication, Dec. 2013.

International Conference Papers

1. B. Defraene, T. van Waterschoot, H.J. Ferreau, M. Diehl, and M. Moonen, “Perception-based clipping of audio signals,” in *Proc. of the 18th European Signal Process. Conf. (EUSIPCO ’10)*, Aalborg, Denmark, Aug. 2010, pp. 517–521.

2. B. Defraene, T. van Waterschoot, M. Diehl, and M. Moonen, “A fast projected gradient optimization method for real-time perception-based clipping of audio signals,” in *Proc. 2011 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP ’11)*, Prague, Czech Republic, May 2011, pp. 333–336.

3. B. Defraene, K. Ngo, T. van Waterschoot, M. Diehl, and M.Moonen, “A psychoacoustically motivated speech distortion weighted multi-channel Wiener filter for noise reduction,” in *Proc. 2012 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP ’12)*, Kyoto, Japan, Mar. 2012, pp. 4637–4640.

4. B. Defraene, T. van Waterschoot, M. Diehl, and M. Moonen, “Perception-based nonlinear loudspeaker compensation through embedded convex optimization,” in *Proc. of the 20th European Signal Process. Conf. (EUSIPCO ’12)*, Bucharest, Romania, Aug. 2012, pp. 2472–2476.

5. T. van Waterschoot, B. Defraene, M. Diehl, and M. Moonen, “Embedded optimization algorithms for multi-microphone dereverberation,” in *Proc. of the 21st European Signal Process. Conf. (EUSIPCO ’13)*, Marrakech, Morocco, Sept. 2013.

6. B. Defraene, T. van Waterschoot, M. Diehl, and M. Moonen, “Embedded-optimization-based loudspeaker compensation using a generic Hammerstein loudspeaker model,” in *Proc. of the 21st European Signal Process. Conf. (EUSIPCO ’13)*, Marrakech, Morocco, Sept. 2013.

International Conference Abstracts

1. B. Defraene, T. van Waterschoot, M. Diehl, and M. Moonen, “Real-time perception-based clipping of audio signals using a fast projected gradient optimization method,” in *Abstracts OPTEC Workshop on Large Scale Convex Quadratic Programming*, Leuven, Belgium, Nov. 2010, pp. 7–8.

National Conference Abstracts

1. B. Defraene, T. van Waterschoot, M. Moonen, H.J. Ferreau, and M. Diehl, "Perceptually optimal clipping of audio signals," in *Abstracts IAP VI/4-DYSCO Study Day*, Leuven, Belgium, Nov. 2009, p. 37.

2. B. Defraene, T. van Waterschoot, M. Diehl, and M. Moonen, "Nonlinear loudspeaker compensation through embedded convex optimization," in *Abstracts IAP VII/19-DYSCO Study Day*, Court-Saint-Etienne, Belgium, Oct. 2012, p. 20.

Technical Reports

1. B. Defraene, T. van Waterschoot, M. Diehl, and M. Moonen, "An ideal ternary time-frequency mask for single-channel speech enhancement," KU Leuven, Leuven, Belgium, ESAT-STADIUS TR 12-89, May 2012.

Curriculum Vitae

Bruno Defraene was born in Halle, Belgium, in 1986. He received the B.Sc. and M.Sc. degree (cum laude) in electrical engineering from KU Leuven, Belgium, in 2007 and 2009, respectively. From 2009, he has worked on his PhD research project at the Electrical Engineering Department of KU Leuven, under the supervision of Prof. Marc Moonen, Prof. Moritz Diehl and the co-supervision of Prof. Toon van Waterschoot.

From 2009 to 2013, he has been a Research Assistant at the Electrical Engineering Department of KU Leuven, and a member of KU Leuven's Optimization in Engineering Center (OPTEC). During this time, he has been the daily supervisor and assessor of 8 M.Sc. theses on diverse audio signal processing topics. He is an active member of the IEEE Signal Processing Society (IEEE SPS) and the European Association for Signal Processing (EURASIP) since 2010. His research interests include audio signal processing, acoustical signal enhancement, audio quality assessment, and optimization for signal processing applications.